

# Aplicação Web Progressiva Aplicada ao Problema de Responsividade e Operacionalidade em Dispositivos Computacionais

Altair Jussadir da Silva Pinto



CENTRO DE INFORMÁTICA  
UNIVERSIDADE FEDERAL DA PARAÍBA

João Pessoa, 2017



Altair Jussadir da Silva Pinto

# Aplicação Web Progressiva Aplicada ao Problema de Responsividade e Operacionalidade em Dispositivos Computacionais

Monografia apresentada ao curso Engenharia de Computação  
do Centro de Informática, da Universidade Federal da Paraíba,  
como requisito para a obtenção do grau de Bacharel em título

Orientador: Prof. Dr. Alisson Vasconcelos de Brito

Junho de 2017

Ficha Catalográfica elaborada por  
Rogério Ferreira Marques CRB15/690

P659a      Pinto, Altair Jussadir da Silva.  
Aplicação web progressiva aplicada ao problema de responsividade e  
operacionalidade em dispositivos computacionais / Altair Jussadir da  
Silva Pinto. – João Pessoa, 2017.  
79p. : il.

Monografia (Bacharelado em Engenharia de Computação) –  
Universidade Federal da Paraíba - UFPB.  
Orientador: Profº. Dr. Alisson Vasconcelos de Brito.

1. Linguagens de programação. 2. Aplicação web progressiva. 3.  
Responsividade. 4. Multiplataforma. I. Título.

UFPB/BSCI

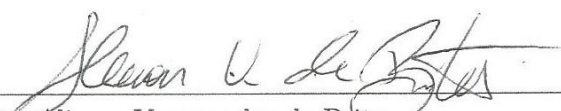
CDU: 004.43 (043.2)



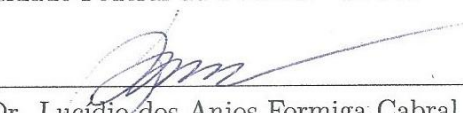


CENTRO DE INFORMÁTICA  
UNIVERSIDADE FEDERAL DA PARAÍBA

Trabalho de Conclusão de Curso de Engenharia de Computação intitulado ***Aplicação Web Progressiva Aplicada ao Problema de Responsividade e Operacionalidade em Dispositivos Computacionais*** de autoria de Altair Jussadir da Silva Pinto, aprovada pela banca examinadora constituída pelos seguintes professores:

  
\_\_\_\_\_  
Prof. Dr. Alisson Vasconcelos de Brito  
Universidade Federal da Paraíba - UFPB

\_\_\_\_\_  
Prof. Dr. Thaís Gaudencio do Rêgo  
Universidade Federal da Paraíba - UFPB

  
\_\_\_\_\_  
Prof. Dr. Lucídio dos Anjos Formiga Cabral  
Universidade Federal da Paraíba - UFPB

\_\_\_\_\_  
Coordenador(a) do Departamento Departamento de Informática  
Thaís Gaudencio do Rêgo  
CI/UFPB

João Pessoa, 14 de Junho de 2017

Centro de Informática, Universidade Federal da Paraíba  
Rua dos Escoteiros, Mangabeira VII, João Pessoa, Paraíba, Brasil CEP: 58058-600  
Fone: +55 (83) 3216 7093 / Fax: +55 (83) 3216 7117



## **DEDICATÓRIA**

Dedico aos meus pais, demais familiares e amigos.

## **AGRADECIMENTOS**

A conclusão da minha graduação não poderia ter sido concretizada sem o auxílio dos meus pais Kátia e Jussadir.

Agradeço também ao professor Alisson Brito pela orientação deste trabalho, e Dangellys Cavalcanti pela orientação de estágio.

Por fim, agradeço aos demais familiares, amigos que são parte da família também e a minha namorada.

## RESUMO

Em diversos sistemas operacionais e dispositivos, costuma-se utilizar diferentes formas de implementação para uma mesma aplicação, porém, com as devidas ferramentas, o funcionamento de uma única aplicação nestes dispositivos torna-se viável, provendo bom desempenho e as funcionalidades da aplicação. A Aplicação Web Progressiva (PWA) é utilizada para poupar que uma mesma aplicação seja desenvolvida para diferentes plataformas, garantindo o mesmo funcionamento independentemente de sistema operacional, e com o auxílio de ferramentas, a auto-adaptação de resolução trazida através da responsividade. O desenvolvimento desse tipo de aplicação pode ser feito de várias formas, optando por uma aplicação nativa ou híbrida. A partir disso será desenvolvido então o Híbrido, uma aplicação híbrida voltada para o mercado comercial virtual, objetivando o estudo sobre esse tipo de tecnologia, abordando principalmente sua forma de resolver problemas voltados a responsividade e operacionalidade.

**Palavras-chave:** *Aplicação Web Progressiva, Responsividade, Multiplataforma.*

## ABSTRACT

Different operating systems and devices use different forms of implementation for the same application, but with the right tools, the operation of a single application on these devices becomes viable, providing good performance and application functionalities. The Progressive Web Application (PWA) is used to save that the same application is developed for different platforms, guaranteeing the same operation independent of operating system, and with the help of tools, the self-adaptation of resolution brought through responsiveness. The development of this type of application can be done in several ways, opting for a native or hybrid application. From this will be developed the Híbrido, a hybrid application focused on the virtual commercial market, aiming the study on this type of technology, addressing mainly its way of solving problems focused on responsiveness and operability.

**Key-words:** Progressive Web Application, Responsiveness, Multiplatform.

## LISTA DE FIGURAS

1	Usuários e aplicativos dos SOs Android, iOS e Windows Phone . . . . .	22
2	Diferença entre acesso via dispositivos computacionais . . . . .	23
3	Resoluções dos dispositivos . . . . .	24
4	Divisão de tela no Bootstrap exemplificando a divisão por <i>grids</i> . . . . .	25
5	Hello World do Web Starter Kit . . . . .	27
6	<i>Dashboard</i> usando Material Design . . . . .	28
7	Cabeçalho do Hibridus . . . . .	30
8	Menu do Hibridus . . . . .	32
9	Página Inicial do Hibridus . . . . .	36
10	Categorias do Hibridus . . . . .	37
11	Carrinho do Hibridus . . . . .	38
12	Informações do Cartão . . . . .	39
13	Armazenamento de Localização . . . . .	40
14	Resultado Galaxy S5 . . . . .	42
15	Resultado Nexus 5X e 6P . . . . .	42
16	Resultado iPhone 5 . . . . .	43
17	Resultado iPhone 6 . . . . .	43
18	Resultado iPhone 6 Plus . . . . .	44
19	Resultado Desktop 1 . . . . .	44
20	Resultado Desktop 2 . . . . .	45
21	Resultado Informações do cartão e Localização no iPhone 6 Plus . . . . .	45
22	Resultado Carrinho no iPhone 6 Plus . . . . .	46
23	Resultado de busca por produto "camisa" . . . . .	47
24	Produto adicionado ao carrinho . . . . .	47
25	Item adicionado ao carrinho . . . . .	48
26	Validação de compra . . . . .	48
27	Compra confirmada . . . . .	49
28	Verificação de informações da compra . . . . .	49

29	Informações no local da compra . . . . .	50
30	Informações do cartão . . . . .	50



## LISTA DE TABELAS

1	Uso de navegadores em dispositivos móveis . . . . .	26
2	Resoluções disponíveis para teste . . . . .	41

## LISTA DE ABREVIATURAS

*PWA - Progressive Web App*

*HTML - HyperText Markup Language*

*CSS - Cascading Style Sheets*

*AJAX - Asynchronous Javascript And XML*

*OS - Operating System*

*GUI - Graphical User Interface*

*HTTPS - Hyper Text Transfer Protocol Secure*

*API - Application Programming Interface*

*IDE - Integrated Development Environment*

## Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>17</b>
1.1	Definição do Problema . . . . .	17
1.2	Premissas e Hipóteses . . . . .	17
1.3	Objetivo geral . . . . .	17
1.4	Objetivos específicos . . . . .	18
1.5	Estrutura da monografia . . . . .	18
<b>2</b>	<b>CONCEITOS GERAIS E REVISÃO DA LITERATURA</b>	<b>19</b>
2.1	Aplicativo . . . . .	19
2.2	Navegador Web . . . . .	19
2.3	Sistema Operacional . . . . .	20
2.4	Responsividade . . . . .	20
2.5	Operacionalidade . . . . .	21
<b>3</b>	<b>METODOLOGIA</b>	<b>22</b>
3.1	Estudo do caso . . . . .	22
3.2	Proposta de Solução . . . . .	26
3.3	Escolhas Tecnológicas . . . . .	27
<b>4</b>	<b>APRESENTAÇÃO E ANÁLISE DOS RESULTADOS</b>	<b>29</b>
4.1	Modelagem . . . . .	29
4.1.1	Barra de navegação . . . . .	29
4.1.2	Menu . . . . .	30
4.1.3	Página inicial . . . . .	32
4.1.4	Categorias . . . . .	36
4.1.5	Carrinho de compras . . . . .	37
4.1.6	Informações do Cartão . . . . .	38
4.1.7	Localização . . . . .	39
4.2	Resultados . . . . .	40

<b>5 CONCLUSÕES E TRABALHOS FUTUROS</b>	<b>51</b>
<b>REFERÊNCIAS</b>	<b>51</b>
<b>APÊNDICE A</b>	<b>54</b>
5.1 Funcionalidades da barra de navegação . . . . .	54
5.2 Implementação do processo de compra . . . . .	56
5.3 Informações de compra . . . . .	69

# 1 INTRODUÇÃO

Os aplicativos são ferramentas computacionais voltados a trazerem soluções, sejam elas simples como para problemas de realizar uma simples conta, como consultar fontes de pesquisa para ajudar o usuário a encontrar o que está procurando, conectar usuários entre si, efetuar compras em lojas, enfim, infinitas possibilidades. Existem os aplicativos nativos, que dependem do sistema operacional do dispositivo para serem executados pois é exigida a instalação do mesmo, e os híbridos, os quais independem de sistema operacional pois sua execução é feita através de navegadores web.

O uso de aplicativos nativos provê o exaustivo trabalho de criação de versões devido a dependência direta do sistema operacional utilizado no dispositivo. Por problemas de incompatibilidade do sistema, essas versões podem vir com vários problemas, tornando a aplicação inoperável. Aplicações híbridas independem de sistema operacional mas sim de navegadores *web*, e a maneira que operam independem de requisitos de hardware ou software, sendo assim uma solução para esse tipo de problema.

## 1.1 Definição do Problema

A adaptação do layout do aplicativo varia de acordo com o dispositivo que está sendo utilizado, sendo assim necessária a criação de várias versões para os diversos dispositivos que o executarão em diferentes sistemas operacionais. Para alguns dispositivos não é feita a devida adaptação, o que compromete parte do seu funcionamento além de ser esteticamente irregular. Esse problema é desencadeado principalmente pela exaustiva produção das versões para diversos dispositivos de variados sistemas operacionais, resultando em alto custo financeiro de retrabalho além de comprometer o lançamento de atualizações para cada dispositivo e sistema operacional vinculado.

## 1.2 Premissas e Hipóteses

Partimos da premissa de que, a adaptação visual e a operacionalidade correta tornam o aplicativo funcional. É possível garantir isso utilizando a propriedade de responsividade e multi operacionalidade.

## 1.3 Objetivo geral

Facilitar a garantia de funcionamento correto de aplicações independentemente de sistema operacional ou resolução de tela do mesmo utilizando a responsividade e a multi operacionalidade providos pelos aplicativos web progressivos.

## **1.4 Objetivos específicos**

Desenvolver uma Aplicação Web Progressiva (do inglês, Progressive Web App - PWA) selecionando ferramentas e implementando uma solução para os problemas de responsividade e operacionalidade.

## **1.5 Estrutura da monografia**

No próximo capítulo, será exibido o problema, proposta uma solução e apresentação das tecnologias a serem utilizadas. Em seguida será apresentado e analisado o resultado obtido a partir da modelagem da aplicação. Por fim, serão feitas as considerações finais e perspectivas para trabalhos futuros.

## 2 CONCEITOS GERAIS E REVISÃO DA LITERATURA

Neste capítulo, serão apresentados os conceitos gerais para nivelamento de conhecimento voltado a área em questão, ou seja, os principais elementos que envolvem este trabalho.

### 2.1 Aplicativo

”“App” pode ser a sigla para várias coisas no mundo, mas em se tratando do mundo *mobile* app é uma abreviação para “application”, do inglês, que significa aplicativo, programa, software. Esta definição simples acaba por ser muito genérica se consideramos que existem apps para celulares, tablets, navegadores e até mesmo sistemas operacionais de desktop.” (BENFICA, 2012).

Segundo Alex Benfica (2012), ao se referir às apps o mais comum é que o assunto seja relacionado aos vários softwares e programas disponíveis para sistemas operacionais móveis. Esse tipo de pensamento se dá devido a maneira com que as pessoas associam um aplicativo ser exclusivamente de um dispositivo móvel.

Porém, softwares computacionais específicos também podem ser chamados de aplicativos baseado em um nome atribuído para um programa computacional voltado para solução de problemas. As soluções podem ser providas independente da escala de complexidade do problema, pois o desenvolvedor do mesmo pode optar pelo tipo de aplicativo, linguagem de programação, dispositivos compatíveis e finalidade, trazendo assim um filtro para que não comprometa o funcionamento da sua solução.

Os aplicativos vão desde simples calendários em um *smartphone*, além de um sistema de comércio virtual ou grandes acervos de livros. Trazem soluções para diversos problemas e podem estar presentes em diversos dispositivos computacionais, até mesmo em relógios inteligentes. Podem ser baixados e instalados através de lojas específicas como Play Store e Apple Store, ou apenas serem executados direto de navegadores web como Google Chrome, Mozilla Firefox, Safari, etc. Os sistemas operacionais mais comuns de apps são Android e iOS.

### 2.2 Navegador Web

”Também conhecidos como *web browsers* ou, simplesmente, *browsers*, os navegadores são uma espécie de ponte entre o usuário e o conteúdo virtual da Internet.” (BOZZA, 2013).

De acordo com Cláudia Bozza (2013), ”os navegadores transformam as páginas codificadas em HTML para uma visualização compreensível para o usuário comum”, ou

seja, o papel deles é transmitir a informação necessária interpretando códigos.

De 1990 até hoje, houve um grande avanço nas páginas web, a principal delas foi o uso de arquivos CSS para formatação da página, definição de aspectos visuais robustos, e o principal elemento responsável pela responsividade destas páginas. Os navegadores processam linguagens de programação como JavaScript, criando assim funcionalidades para o usuário, desde uma submissão de formulário até criar telas de carregamento.

Um dos grandes avanços nos últimos anos foi o uso da biblioteca React desenvolvida pelos desenvolvedores do Facebook. "Toda vez que um elemento precisa ser renderizado, a biblioteca identifica as atualizações comparando o elemento no DOM e na memória, modificando somente o que foi de fato alterado, sem precisar renderizar todo o elemento." (FERREIRA, 2013), ou seja, reduz o consumo de dados para carregamento da página.

## 2.3 Sistema Operacional

"O sistema operacional permite que o usuário execute tarefas como o gerenciamento das aplicações em execução, do acesso à memória e dos arquivos no disco. Além disso, o sistema operacional traduz para o computador comandos dos usuários. Por exemplo, quando o usuário clica duas vezes sobre um ícone, o comando é traduzido pelo sistema operacional para o computador "entender" que o usuário deseja abrir um programa." (BOZZA, 2013).

Existem vários sistemas operacionais, porém os mais consolidados no ramo da informática são: Microsoft Windows, Linux e Mac OS X. Através de uma interface gráfica (GUI), sua principal função é levar ao usuário o que o computador entende através de suas ações, exibindo apenas o que é de interesse do mesmo, ou seja, permite que o usuário dê ordens ao computador e ele obedeça seus comandos. O sistema operacional mais comum em computadores é o Microsoft Windows que tem versões desde 1980 até hoje.

Além destes OSs, foram desenvolvidos os sistemas operacionais móveis e atribuídos a esse tipo de dispositivos como Android, iOS, Windows Phone e Symbian, sendo destes os mais comuns Android e iOS. Do lado do Android temos marcas como Motorola, Lenovo, Samsung, Alcatel, etc. tendo diferentes versões entre si, pois cada versão depende do hardware que o dispositivo possui para que o seu OS seja atualizado. Por outro lado, temos o iOS, disponível apenas em dispositivos da Apple que, quando descontinuados, não atualizam mais os OSs.

## 2.4 Responsividade

"Design Responsivo é uma técnica de estruturação HTML e CSS, que consiste em adaptar o site ao *browser* do usuário sem que seja necessário definir várias folhas de estilos



específicas para cada resolução, ou seja, é um tipo de design onde o *layout* fica fluído e variante de acordo com a resolução do usuário.”(ARRIGONI, 2015).

Tornar o design responsivo ajuda o usuário a ver corretamente as informações, seja num site ou num aplicativo. Isso está totalmente ligado a questão do funcionamento correto da aplicação, pois quando o design se adapta a tela do usuário de maneira esperada pelo desenvolvedor, todos os elementos da página ficam acessíveis, provendo total interação e funcionamento.

Garantir que o *layout* da aplicação ocupe adequadamente o *display* do dispositivo móvel vai além de uma questão de beleza visual, entra na parte de funcionamento. Através de algumas preocupações do desenvolvedor, ele pode garantir que não será necessário retrabalho para outro tipo de resolução, assim como a fluidez do seu aplicativo desde *desktops* até *smartphones*.

## 2.5 Operacionalidade

Segundo Fernando Manuel (2011), operacionalidade é ”Qualidade de o que é operável. Permite-se através de diversas operações manter sua utilidade. Perante exigências de qualquer ordem mantém a sua racionalidade.”.

No caso dos aplicativos, o que vai garantir essa qualidade é a maneira como o software será manipulado, a correta implementação das funcionalidades e como essa aplicação se ligará ao hardware que está sendo utilizado para sua execução.

Utilizando da responsividade e a correta implementação do PWA, garantimos que o sistema funcionará corretamente, se adaptando a diversos dispositivos em diferentes resoluções.

### 3 METODOLOGIA

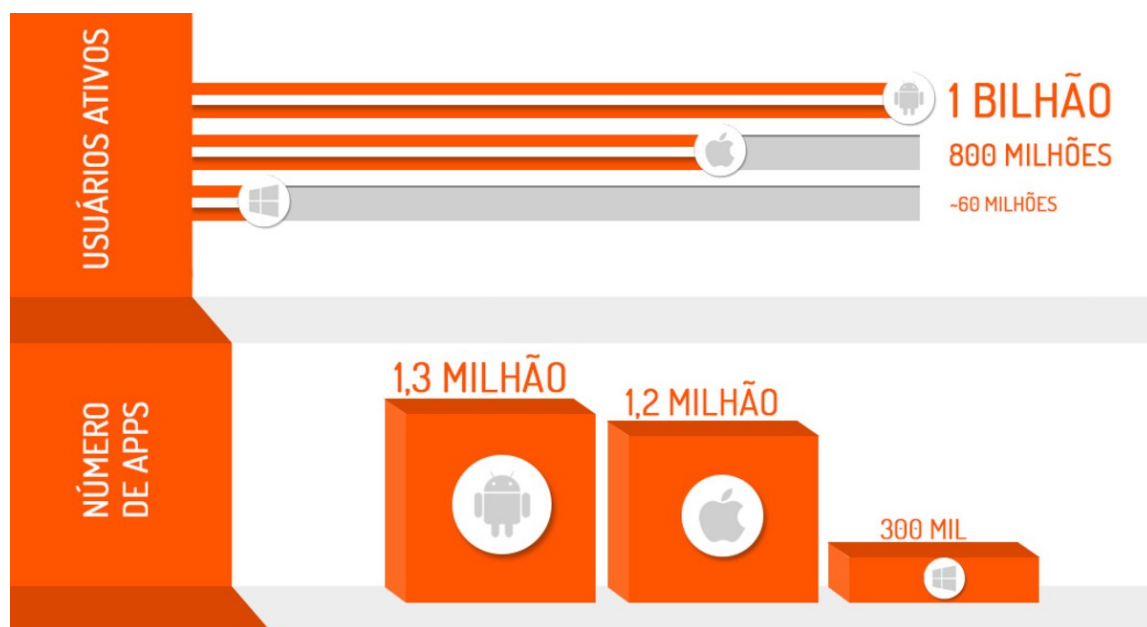
Neste capítulo será discutida a metodologia escolhida para a resolução dos problemas em questão. A avaliação de eficiência será baseada em: adaptação visual ao *display* dos dispositivos, garantindo que nenhum elemento será sobreposto ou posicionado fora do espaço destinado para o mesmo, e capacidade de usabilidade total do sistema, garantindo que o usuário seja capaz de realizar todas as operações providas.

#### 3.1 Estudo do caso

Aplicativos móveis são cada vez mais requisitados no mercado, e de acordo com a demanda, vem a oferta. Isso faz com que os desenvolvedores se preocupem cada vez mais em como seu aplicativo irá se comportar em determinado dispositivo, tanto na parte de execução correta de suas funcionalidades como a de adaptação de *layout*.

Os tipos mais comuns são os aplicativos nativos, os quais necessitam de instalação e um específico sistema operacional para executá-lo. A **Figura 1** mostra o número de usuários ativos dos sistemas operacionais *mobile* mais utilizados e o número de aplicativos desse tipo desenvolvidos para o mesmo.

**Figura 1: Usuários e aplicativos dos SOs Android, iOS e Windows Phone**



Fonte: <https://www.tecmundo.com.br/sistema-operacional/60596-ios-android-windows-phone-numeros-gigantes-comparados-infografico.htm>

Baseado nisso, temos então uma visão geral da quantidade de versões que devem ser desenvolvidas para cada sistema operacional, versão do SO e dispositivo que o utiliza. Isso gera um problema de adaptação e consequentemente funcionamento correto de uma

aplicação caso o desenvolvedor não volte sua atenção especificamente para dado dispositivo e versão do SO, podendo assim criar uma versão incompatível, o que gera todo o custo de lançamento de atualização tanto para ele como para os usuários.

Atrelado a isso, vem o problema de adaptação de *layout*. Na condição nativa ela se dá de acordo com o dispositivo em questão e sistema operacional. Numa visão mais abrangente, temos as aplicações híbridas, ou aplicações web. Embora voltadas principalmente para dispositivos móveis, enfrenta a dificuldade de se adaptar ao *layout* da tela do dispositivo em questão.

A técnica de design responsivo originou-se em 2010 inicialmente para melhorar a usabilidade de páginas web e com o passar do tempo foi se difundindo para sistemas mais complexos. Tornou-se então necessária a adaptação para os dispositivos computacionais que surgiam com os avanços tecnológicos como smartphones, tablets, netbooks, ultrabooks, etc.

Baseado nessa necessidade, a preocupação dos desenvolvedores ficou cada vez maior em exibir corretamente seus sites nos diversos dispositivos, pois o acesso deixou de ser apenas feito em desktops com grandes monitores de escalas praticamente únicas e partiu para telas de pouco mais de 4 polegadas. Na **Figura 2** podemos ver a diferença de acesso a sites em desktops e dispositivos móveis em 2010 e 2012.

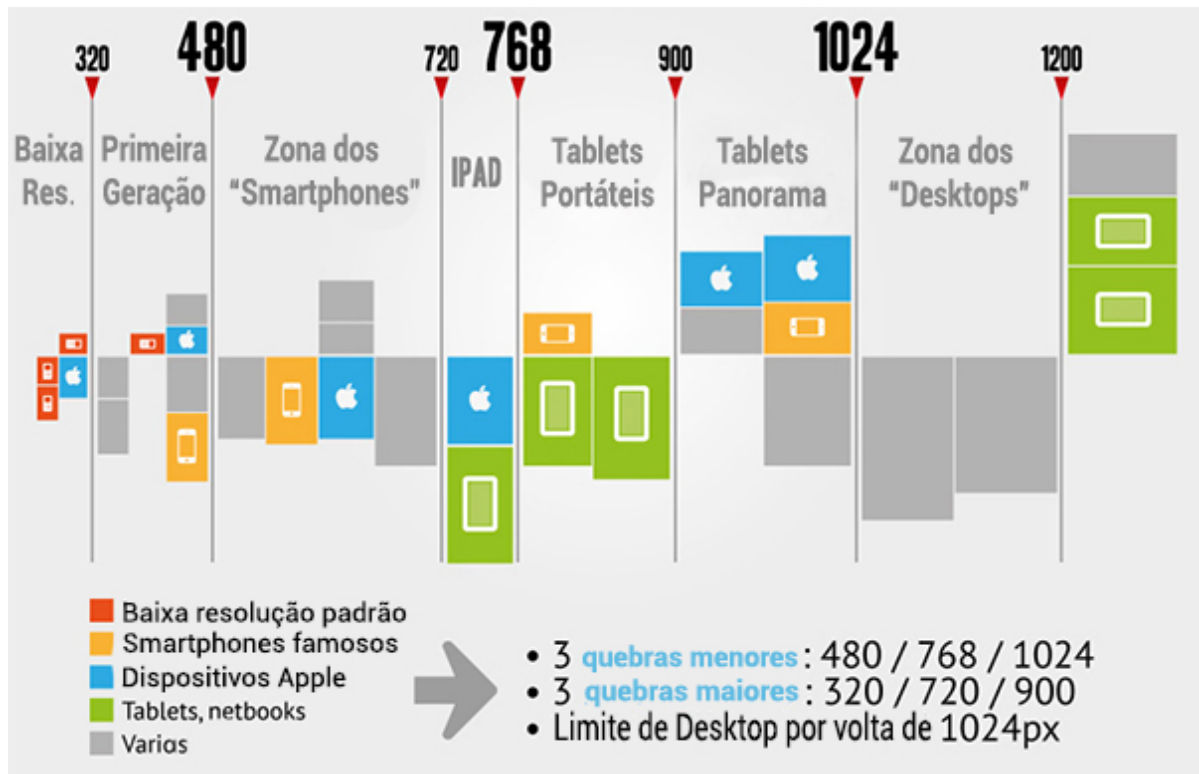
**Figura 2: Diferença entre acesso via dispositivos computacionais**



Fonte: <http://www.devmedia.com.br/responsive-design-dicas-para-tornar-seu-site-acessivel-em-qualquer-resolucao/28316>

É importante também conhecer as resoluções dos dispositivos que se tem interesse em desenvolver sistemas para ele. Na **Figura 3** vemos uma escala em pixel das principais resoluções utilizadas no mercado de dispositivos computacionais.

Figura 3: Resoluções dos dispositivos



Fonte: <http://www.devmedia.com.br/responsive-design-dicas-para-tornar-seu-site-acessivel-em-qualquer-resolucao/28316>

Para aplicação de responsividade, existem alguns *frameworks* para resolver esse problema pro desenvolvedor, sem se preocupar em como deve organizar os elementos na tela para que se comporte de maneira correta nessas diferentes resoluções. Estes *frameworks* dividem a tela em *grids* e, a partir disso, o desenvolvedor pode inserir elementos em determinadas posições que se adaptam as resoluções.

Um dos mais famosos *frameworks* para isso é o Bootstrap. Ele se baseia em *grids* que servem para orientar o tamanho e posição de objetos na tela, como mostra a **Figura 4**, onde vemos como são feitos os posicionamentos em dispositivos como smartphones, tablets e desktop.

**Figura 4: Divisão de tela no Bootstrap exemplificando a divisão por *grids***

#### Mixed: mobile and desktop

The Bootstrap 3 grid system has four tiers of classes: xs (phones), sm (tablets), md (desktops), and lg (larger desktops). You can use nearly any combination of these classes to create more dynamic and flexible layouts.

Each tier of classes scales up, meaning if you plan on setting the same widths for xs and sm, you only need to specify xs.

.col-xs-12 .col-md-8		.col-xs-6 .col-md-4	
.col-xs-6 .col-md-4	.col-xs-6 .col-md-4		.col-xs-6 .col-md-4
.col-xs-6		.col-xs-6	

#### Mixed: mobile, tablet, and desktop

.col-xs-12 .col-sm-6 .col-lg-8		.col-xs-6 .col-lg-4	
.col-xs-6 .col-sm-4	.col-xs-6 .col-sm-4		.col-xs-6 .col-sm-4

Fonte: <http://getbootstrap.com/examples/grid/>

Esse tipo de divisão garante que os elementos posicionados na tela não se sobreponham e adaptem-se as resoluções dos dispositivos que utilizarão a aplicação. Não é necessário que sejam *grids* mas que o desenvolvedor garanta que o campo de visão da aplicação se comporte da mesma maneira, podendo ser com espaçamentos ou ocultação de informações irrelevantes.

Garantindo a devida responsividade do nosso sistema, temos o correto funcionamento do mesmo em diversos dispositivos, sem a preocupação de como será adaptado à determinada resolução em dado dispositivo.

Porém, para que se execute corretamente uma aplicação web, deve-se observar a compatibilidade do dispositivo com os navegadores disponíveis, de preferência os que possuem o HTML5. Baseado nisso, são analisados os principais navegadores utilizados em todo o mundo para que, focado mais em um determinado navegador, se tenha uma base de completa operacionalidade da aplicação. A **Tabela 1** mostra o percentual de uso e os respectivos navegadores em dispositivos móveis.

**Tabela 1: Uso de navegadores em dispositivos móveis**

Navegador	Percentual de uso (março de 2017)
Chrome	54,14 %
Safari	14,26 %
UC Browser	8,40 %
Firefox	6,21 %
Internet Explorer	4,03 %
Opera	3,56 %
Outros (total)	9,43 %

Fonte: <http://gs.statcounter.com/>

### 3.2 Proposta de Solução

Com base nas informações anteriores, torna-se necessário utilizar uma ferramenta para solucionar os problemas de execução da aplicação em diversos dispositivos independente de SO em questão, e para a correta adaptação nos mais comuns dispositivos computacionais no mercado atualmente.

A partir disso, é proposto o PWA, que combina o melhor da web e dos aplicativos. O PWA traz com ele as seguintes características:

- Progressivo: Funciona em navegadores web independente do escolhido pelo usuário.
- Responsivo: Se adapta a qualquer formato de tela, desde desktops a smartphones.
- Baixo uso de dados: Utiliza de *serviceworkers* para trabalhar em redes de qualidade baixa ou, em alguns casos, *offline*.
- Seguro: Certificação HTTPS na conexão.
- Acesso através de link: Não precisa de instalação e permite acesso rápido ao seu endereço através de atalhos.

Com essas características, soluciona-se o problema de responsividade e operabilidade dos aplicativos nos mais comuns dispositivos computacionais.

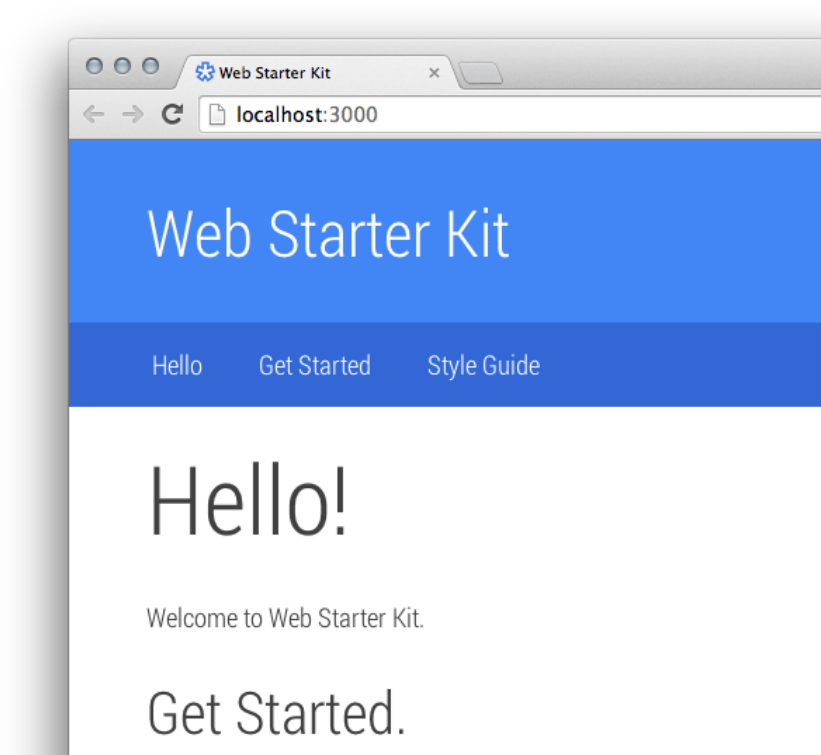
Para a exibição de um caso real de uso do PWA, será criado um sistema de *e-commerce* cujo nome será Híbrido, onde serão expostos alguns produtos a venda e o usuário será capaz de realizar compras.

### 3.3 Escolhas Tecnológicas

Serão adotadas algumas tecnologias para desenvolvimento do PWA. Inicialmente, o uso do Web Starter Kit da Google, uma ferramenta cujo objetivo é simplificar o desenvolvimento web. Embora seja simples como a própria definição sugere, é um ponto de partida para o desenvolvimento web para múltiplos dispositivos.

Acompanha vários pacotes que são instalados nele, como o gulp, que serve para gerar um *socket* e permitir a conexão a partir de um servidor local, em outros dispositivos. Traz consigo um projeto inicial, como mostra a **Figura 5** contendo o endereço local e a página inicial.

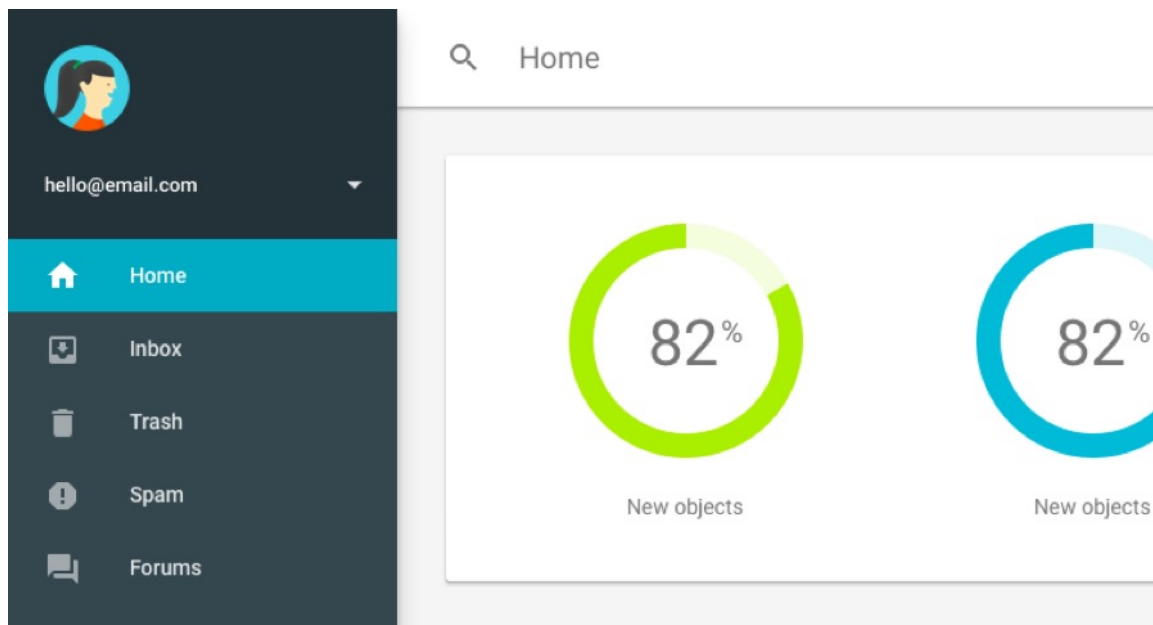
**Figura 5: Hello World do Web Starter Kit**



Fonte: <https://webdesign.tutsplus.com/pt/tutorials/get-up-and-running-with-google-web-starter-kit-cms-21495>

Para a parte visual com adaptação automática aos diferentes *layouts*, será utilizado o Material Design, que trata-se do padrão visual da Google que está presente no SO Android. Contém cores chapadas e vibrantes, fontes personalizadas e ícones, como mostra a **Figura 6**.

**Figura 6:** *Dashboard* usando Material Design



Fonte: <https://getmdl.io/templates/index.html>

Para exibição de janelas modais providos pelo Material Design em todos os navegadores, será utilizado o pacote Dialog Polyfill.



## 4 APRESENTAÇÃO E ANÁLISE DOS RESULTADOS

De acordo com o que foi apresentado nos capítulos 2 e 3, este capítulo apresenta o PWA proposto a fim de mostrar a solução que o mesmo provê para as questões de responsividade e operacionalidade em diversos dispositivos computacionais.

A IDE utilizada para desenvolvimento foi o Visual Studio Code de versão 1.1, sistema operacional Windows 10.1 x64, Notebook Acer Aspire E 15 Core i7-5500U 2.4GHz, NVIDIA GeForce 920M 2GB VRAM, 8GB Ram DDR3 e 1TB HD.

### 4.1 Modelagem

Aqui serão exibidos os elementos modelados para que a aplicação se comporte de acordo com a resolução do dispositivo em questão. Todos os componentes foram baseados ou utilizados do padrão visual da Google e o código utilizado será exibido em seu respectivo elemento a fim de demonstrar como foi garantida a responsividade do mesmo.

#### 4.1.1 Barra de navegação

A barra de navegação é fixa, ou seja, sempre posicionada no canto superior da tela, sem a necessidade de ter sua posição alterada. Para desenvolvimento da mesma, foram atribuídas algumas propriedades do CSS à classe *header* a fim de garantir que, para todas as resoluções possíveis, ela mantenha um comprimento de 40px e largura da tela completa, ou seja, 100%. Ela é fixada no topo a partir da propriedade *position*, que está definida como *fixed*.

```
1  .header {  
2      width: 100%;  
3      height: 40px;  
4      text-align: center;  
5      background: dodgerblue;  
6      position: fixed;  
7  }
```

---

O resultado obtido pode ser visto na **Figura 7**:

Figura 7: Cabeçalho do Hibridus



Fonte: Autor

#### 4.1.2 Menu

Para o menu ser exibido, basta clicar no "hambúrguer" ao lado do nome Hibridus na barra de navegação. Para ser modelado, foi utilizada a classe *navigation* se preocupando em manter uma margem superior de 40px e esquerda de 25px. O comprimento de 470px é suficiente para que sejam exibidos todos os elementos com uma largura máxima de 250px. A sua posição foi definida fixa para que se sobreponha aos demais elementos na tela, bordas arredondadas e fundo com efeito.

Foi utilizada a propriedade de *Media Queries* do CSS para ajustar o menu a determinada resolução. Também foram inseridas propriedades para os elementos do menu baseados no tamanho máximo definido para o mesmo, tudo utilizando valores em %. Também foi adicionado um botão de busca associado a um campo de texto para preenchimento da palavra a ser buscada. A parte do código referente a isso pode ser visto a seguir:

```
1 .navigation {
```

```

2      border-top-left-radius: 0px;
3      border-top-right-radius: 0px;
4      border-bottom-left-radius: 15px;
5      border-bottom-right-radius: 15px;
6      margin-top: 40px;
7      margin-left: 25px;
8      height: 470px;
9      width: 0;
10     max-width: 250px;
11     position: fixed;
12     z-index: 1;
13     top: 0;
14     left: 0;
15     overflow-x: hidden;
16     transition: 0.5s;
17     padding-top: 0px;
18     background: rgb(44,62,80);
19     background-image: linear-gradient(135deg, rgb(3,8,12), rgb
        (16,57,79));
20 }
21
22 @media screen and (max-height: 450px) {
23     .navigation {
24         padding-top: 15px;
25     }
26     .navigation a {
27         font-size: 18px;
28     }
29 }
30
31 #txtBusca{
32     margin-top: 3px;
33     margin-left: 15px;
34     float:left;
35     background-color:transparent;
36     font-style:italic;
37     font-size:14px;
38     border:none;
39     height:30px;
40     width:65%;
41     color: #e7e7e7;
42 }
43
44 #btnBusca{
45     float:right;
46     background-color:transparent;
47     height:28px;

```

```

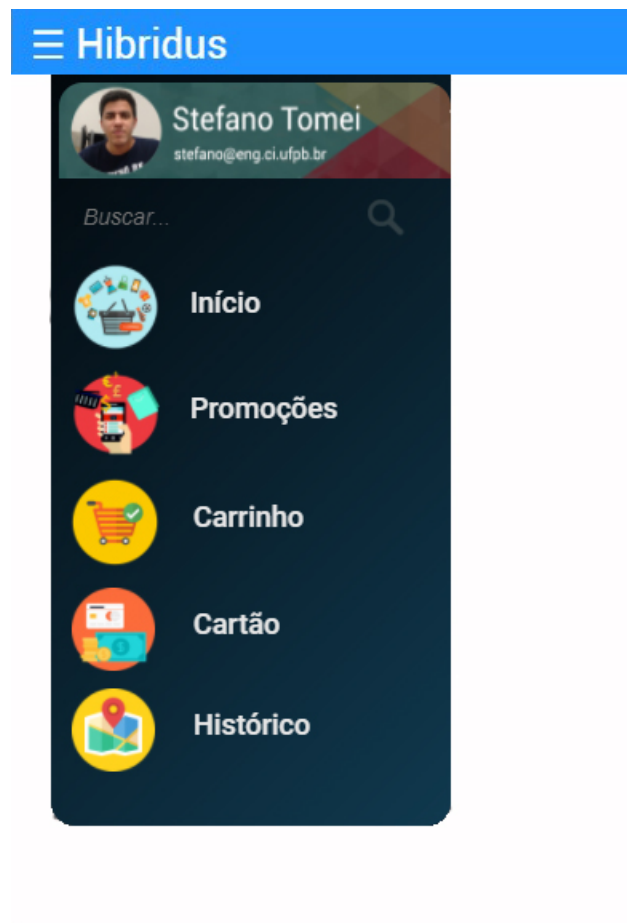
48     color: #e7e7e7;
49 }
50
51 .busca{
52     width: 270px;
53     height: 70px;
54     -webkit-background-size: 100%;
55     -o-background-size: 100%;
56 }

```

---

O resultado obtido pode ser visto na **Figura 8**:

**Figura 8: Menu do Hibridus**



Fonte: Autor

#### 4.1.3 Página inicial

Para a página inicial, foi utilizado o conceito de *container*, onde todos os elementos serão inseridos nele e assim será garantido o comportamento dentro da área de exibição. A classe *container* utiliza toda a largura da tela com um pequeno espaçamento superior para que se posicione o cabeçalho da página sem interferir em seu conteúdo.

Inicialmente, será posicionado abaixo do cabeçalho um menu horizontal para as categorias disponíveis. Para garantir que se comporte de maneira correta na resolução em questão, foi definida uma largura máxima de 1700px, com largura definida para ocupar toda a tela em 100%, garantindo assim que para resoluções pequenas, haverá uma barra de rolagem que exibirá os demais itens ao ser arrastada, definindo o *overflow* como *auto* e escondendo-o. Seu posicionamento deve ser relativo pois deve respeitar o espaço destinado para o cabeçalho.

Haverá também a divisão por linhas de classe denominada *row* e colunas de classes denominadas *col*, *col-2* e *col-3*, responsáveis por organizar elementos numa mesma coluna, independente da resolução pois sua ocupação estará definida como 100%. para apenas uma coluna e 50% para duas colunas. No caso de três colunas, são ignorados os espaçamentos entre as duas colunas.

```
1      .container {
2          width: 100%;
3          text-align: center;
4          margin-left: auto;
5          margin-right: auto;
6          padding: 0 20px;
7      }
8
9      @media (min-width: 1170px) {
10         .container {
11             max-width: 1170px;
12             padding: 0;
13         }
14     }
15
16     .container-card {
17         float: left;
18         margin-top: 10%;
19         margin-left: auto;
20     }
21
22     .col {
23         display: inline-block;
24         vertical-align: top;
25         min-height: 1px;
26         padding-left: 5px;
27         padding-right: 5px;
28         width: 50%;
29         margin-right: -4px;
30         *zoom: 1;
31         *display: inline;
```

```

32     }
33
34     @media (min-width: 480px) {
35         .col-2 {
36             width: 50%;
37         }
38     }
39
40     .col-3 {
41         vertical-align: top;
42         min-height: 1px;
43         padding-left: 7px;
44         padding-right: 5px;
45         width: 100%;
46         margin-right: -4px;
47         *zoom: 1;
48         *display: inline;
49     }
50
51     .scrollmenu {
52         margin-top: 40px;
53         background-color: #333;
54         overflow: auto;
55         white-space: nowrap;
56         max-width: 1700px;
57         width: 100%;
58         position: relative;
59         box-shadow: 3px 2px 2px 2px rgba(0, 0, 0, 0.14), 0 3px 1px
60             -2px rgba(0, 0, 0, 0.2), 0 1px 5px 0 rgba(0, 0, 0, 0.12);

```

---

Então, são inseridos no HTML da seguinte forma:

```

1     <div id="rodape">
2         <div class="scrollmenu" onclick="closeNav()">
3             <a href="clothing.html">Roupas</a>
4             <a href="accessories.html">Acessorios</a>
5             <a href="house.html">Casa</a>
6             <a href="decoration.html">Decoracao</a>
7             <a href="technology.html">Tecnologia</a>
8         </div>
9     </div>
10
11
12     <div class="container" onclick="closeNav()">
13         <div class="row">

```

```

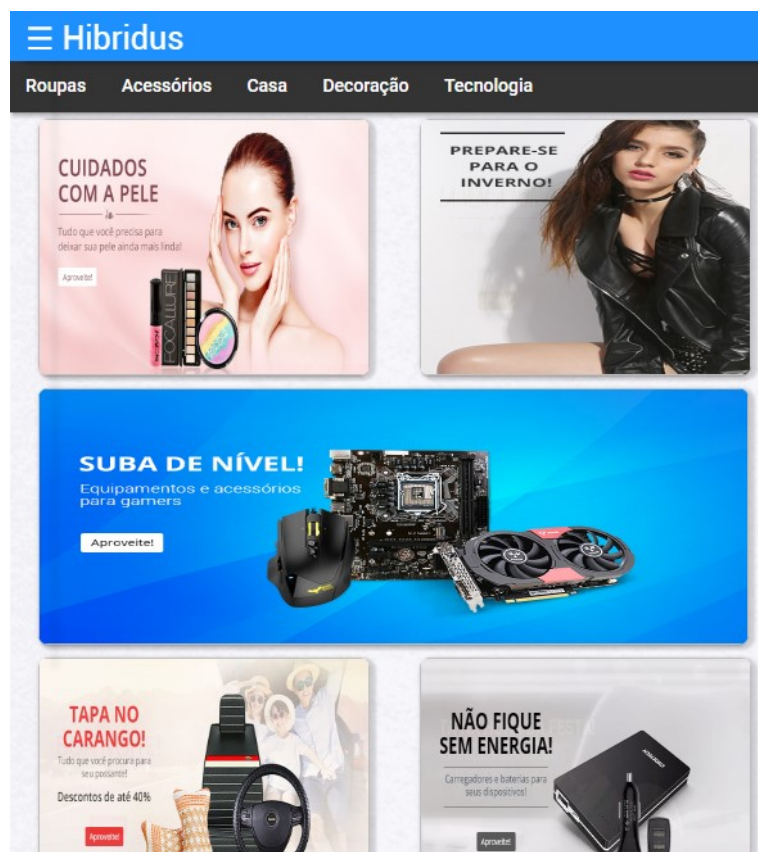
14         <div class="col col-2">
15             <a href="accessories.html"><img class="card_1" /></a>
16         </div>
17         <div class="col col-2">
18             <a href="clothing.html"><img class="card_2"/></a>
19         </div>
20     </div>
21
22     <div class="row">
23         <div class="col-3">
24             <a href="technology.html"><img class="card_3"/></a>
25         </div>
26     </div>
27     <div class="row">
28         <div class="col col-2">
29             <a href="decoration.html"><img class="card_4"/></a>
30         </div>
31         <div class="col col-2">
32             <a href="house.html"><img class="card_5"/></a>
33         </div>
34     </div>
35 </div>
36 </div>

```

---

Todas as imagens atribuídas aos *cards* são adaptáveis ao seu tamanho, sendo definidas como *background* e assim se auto-ajustando. O resultado pode ser visto na **Figura 9** onde embora se tenha mais conteúdo abaixo, a barra de rolagem é ocultada porém exibida ao rolar a página para baixo:

Figura 9: Página Inicial do Híbrido



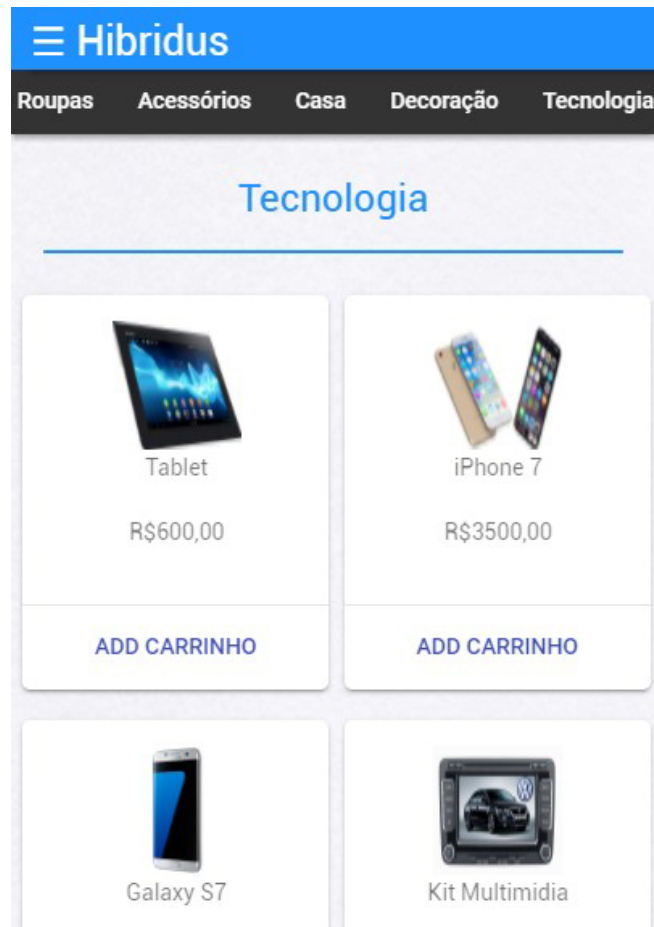
Fonte: Autor

#### 4.1.4 Categorias

As categorias seguem a mesma ideia da página inicial quanto ao posicionamento de *cards* em colunas. Todos são adicionados a um *container* e, a partir disso, se comportam de maneira responsiva na resolução em questão. Pode ser visto então na **Figura 10** o comportamento desses *cards* em apenas uma tela pois as demais são repetitivas, alterando apenas os produtos em questão.



Figura 10: Categorias do Híbrido



Fonte: Autor

#### 4.1.5 Carrinho de compras

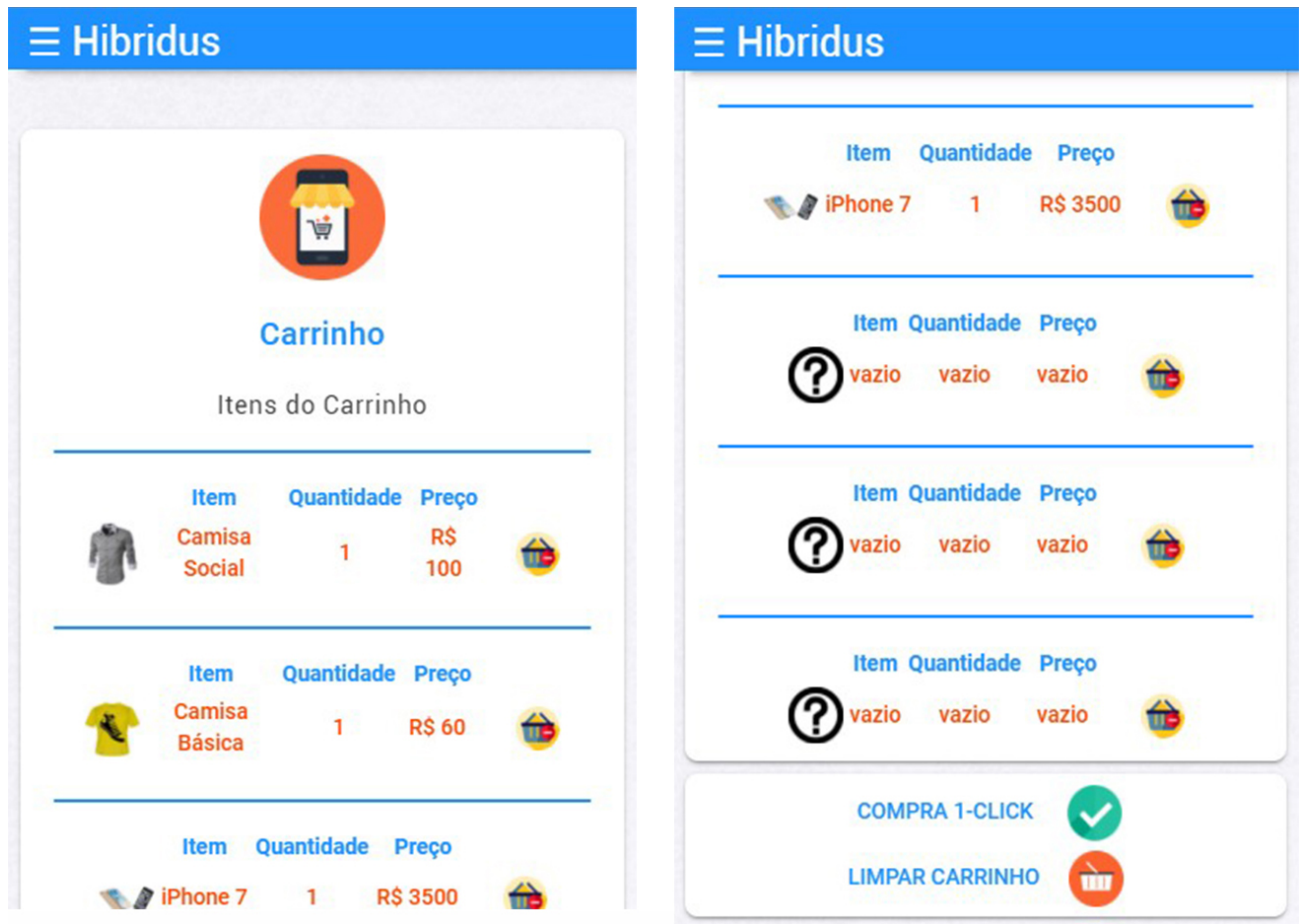
No carrinho é utilizado um grande *card* que envolve toda a parte da tela que terá os respectivos itens adicionados ao mesmo, e por fim as opções em um outro *card* menor, ambos se comportando da mesma maneira de acordo com a resolução, porém, divididos na tela por questões mais estéticas do que funcionais.

Os itens são exibidos em formato de tabela, porém usado um *container* novo chamado *container-2*. As atribuições dos elementos podem ser vistas no código a seguir:

```
1  .container-2 {  
2      width: 100%;  
3      margin-top: 70px;  
4      margin-left: auto;  
5      margin-right: auto;  
6      padding: 0 20px;  
7      text-align: center;  
8  }
```

O resultado pode ser visto na **Figura 11** onde é exibida a respectiva tela dividida em duas partes:

**Figura 11: Carrinho do Hibridus**



Fonte: Autor

#### 4.1.6 Informações do Cartão

A tela contendo informações do cartão segue a mesma base da tela de carrinho. Sua particularidade é como se monta a tabela da mesma.

Um *script* é executado para geração da tabela, porém, ela é modelada no CSS a fim de melhorar sua exibição da seguinte maneira:

```

1  #tabela table {
2      border: 1px #c0c0c0 solid;
3      width: 100%;
4      margin-bottom: 7%;
5  }
6  #tabela table th {
7      background-color: dodgerblue;

```

```

8     color: #fff;
9 }
10 #tabela table td {
11     background-color: #f0f0f0;
12     color: #0c0c0c;
13 }

```

O resultado obtido pode ser visto na **Figura 12**:

**Figura 12: Informações do Cartão**

**Hibridus**

**Número do Cartão**  
XXXX XXXX XXXX 2280

---

**Limite Disponível**  
R\$ 6650,00

---

**Extrato**

Data/Hora	Descrição	Valor
14/02/2017 00:30:56	Camisa Social (Promoção) x 1	R\$ 50
14/02/2017 00:31:09	Camisa Social (Promoção) x 2	R\$ 100
14/02/2017 00:32:51	Camisa Social (Promoção) x 2	R\$ 100
09/06/2017 16:32:00	Camisa Social (Promoção) x 2	R\$ 100

**ENVIAR FATURA POR EMAIL**

Fonte: Autor

#### 4.1.7 Localização

A parte de localização é ainda mais simples, pois o desenvolvedor preocupa-se apenas em definir o tamanho máximo do mapa, que nesse caso é o tamanho do componente *container*.

```

1  #map {
2      height: 90%;
3      width: 90%;

```

```

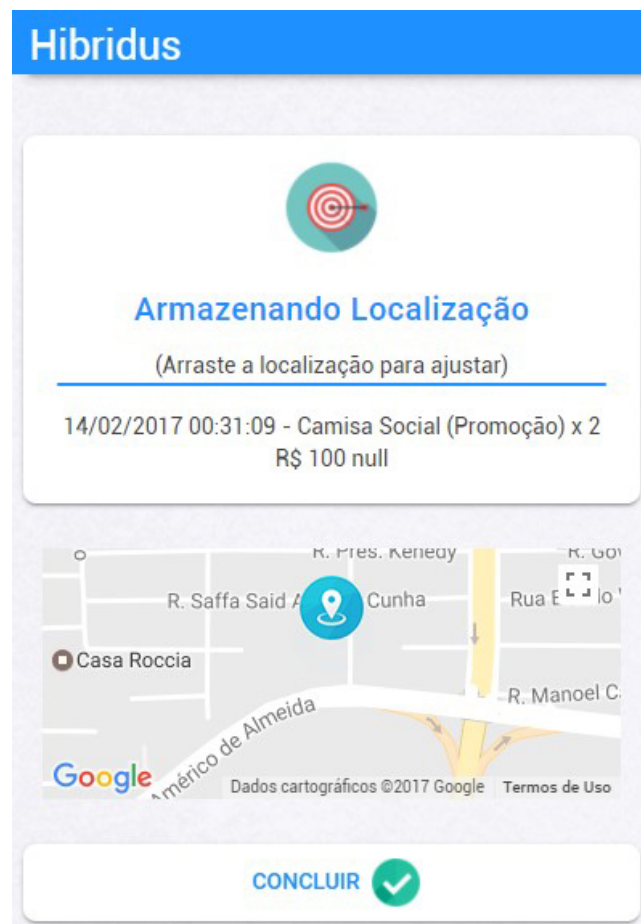
4     margin: 20px;
5 }

```

---

Portanto, na **Figura 13** é possível ver como se comportam os componentes no momento de capturar a localização atual.

**Figura 13: Armazenamento de Localização**



Fonte: Autor

## 4.2 Resultados

Associando o que foi modelado em CSS, após a devida integração com o HTML, são expostos os resultados, resultados que foram exibidos anteriormente nas figuras relacionadas ao Híbridus. Alguns componentes não precisaram ser modelados ou posicionados pois, como dito no começo deste capítulo, foram usados alguns elementos do kit de desenvolvimento da Google.

Para execução da aplicação, foi utilizado o comando *gulpserve* gerando assim um socket para acesso remoto ao servidor local. Foi também utilizado o HTTPS pois o Google Maps necessita do certificado para capturar a localização atual do dispositivo. Como não

havia a chave, a conexão é dada como insegura, porém não apresenta problemas para o usuário tendo em vista que a máquina usada para desenvolvimento era o servidor.

Para demonstrar melhor a eficiência da aplicação de responsividade para correta operacionalidade, serão exibidos os resultados de adaptação de layout em diferentes resoluções. Baseado no console do Google Chrome, podemos observar as seguintes resoluções como mostra a **Tabela 2**:

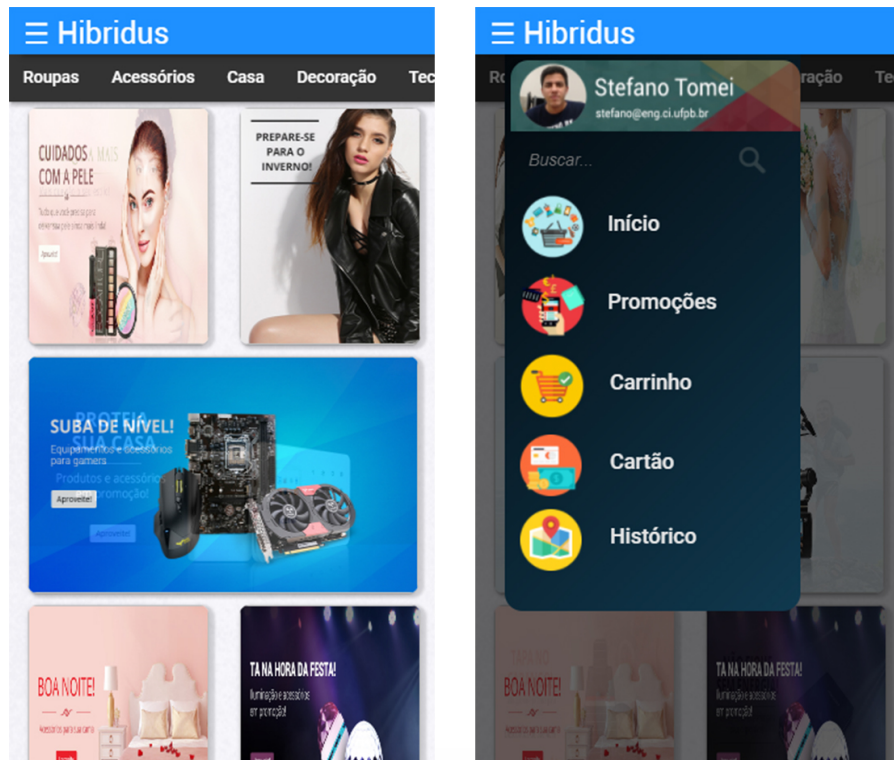
**Tabela 2: Resoluções disponíveis para teste**

Dispositivo	Resolução (largura x altura)
Galaxy S5	360 x 640
Nexus 5X e 6P	412 x 732
iPhone 5	320 x 568
iPhone 6	375 x 667
iPhone 6 Plus	414 x 736
iPad	768 x 1024
iPad Pro	1024 x 1366
Desktop	1366 x 768

Fonte: Google Chrome

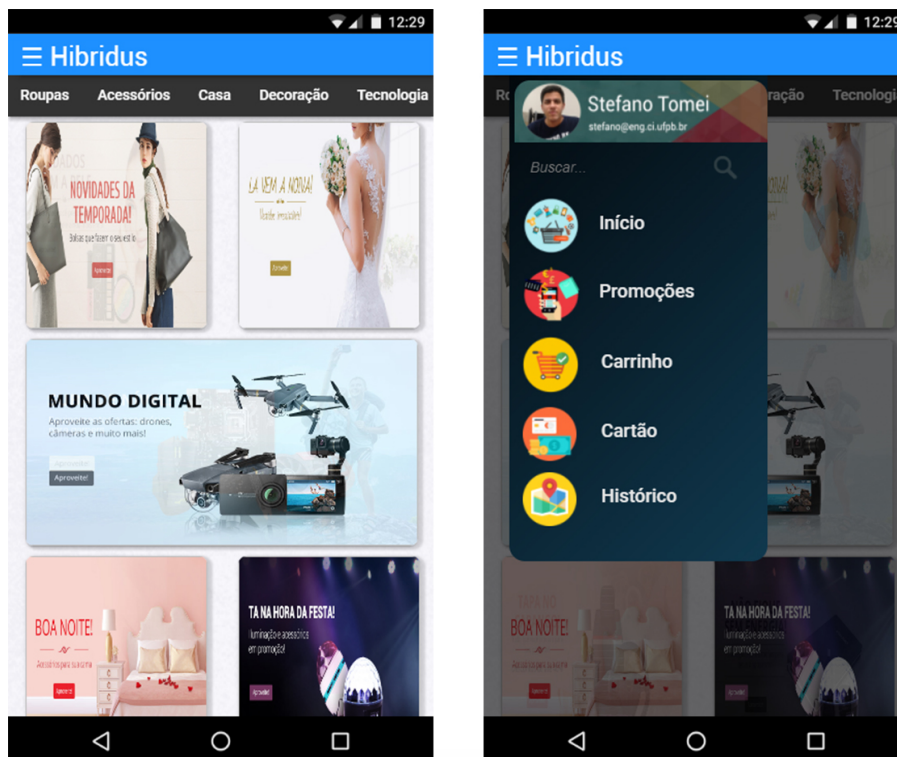
No caso geral, a maioria dos dispositivos atualmente no mercado utilizam dessas resoluções, portanto, serão adotados como testes gerais os experimentos de acordo com essas resoluções. Nas figuras abaixo serão exibidos os resultados de acordo com a resolução e em comparativo também o modelo de adaptação diferente, que foi o do menu. A diferença entre imagens nos cards se dá pelo fato de que eles trocam constantemente o anúncio através de transições do CSS.

Figura 14: Resultado Galaxy S5



Fonte: Autor

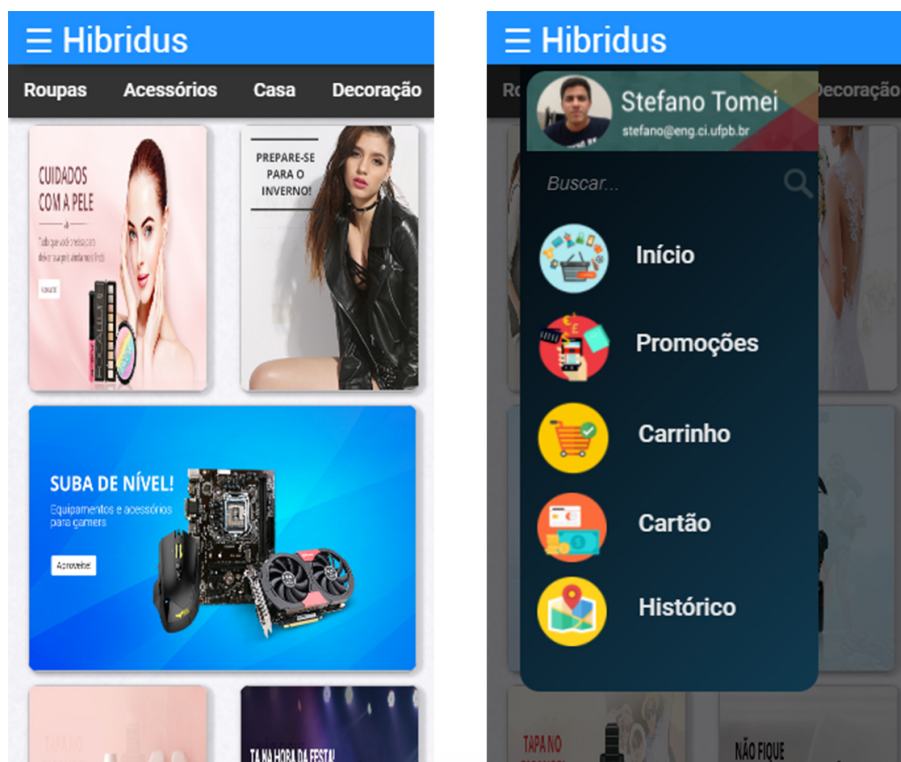
Figura 15: Resultado Nexus 5X e 6P



Fonte: Autor

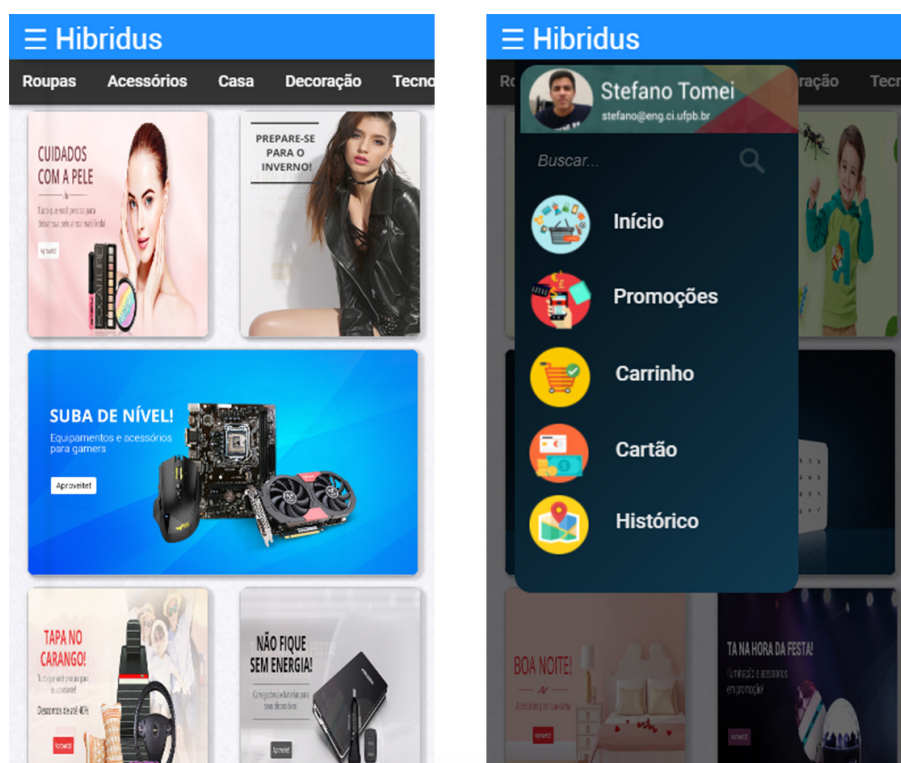


Figura 16: Resultado iPhone 5



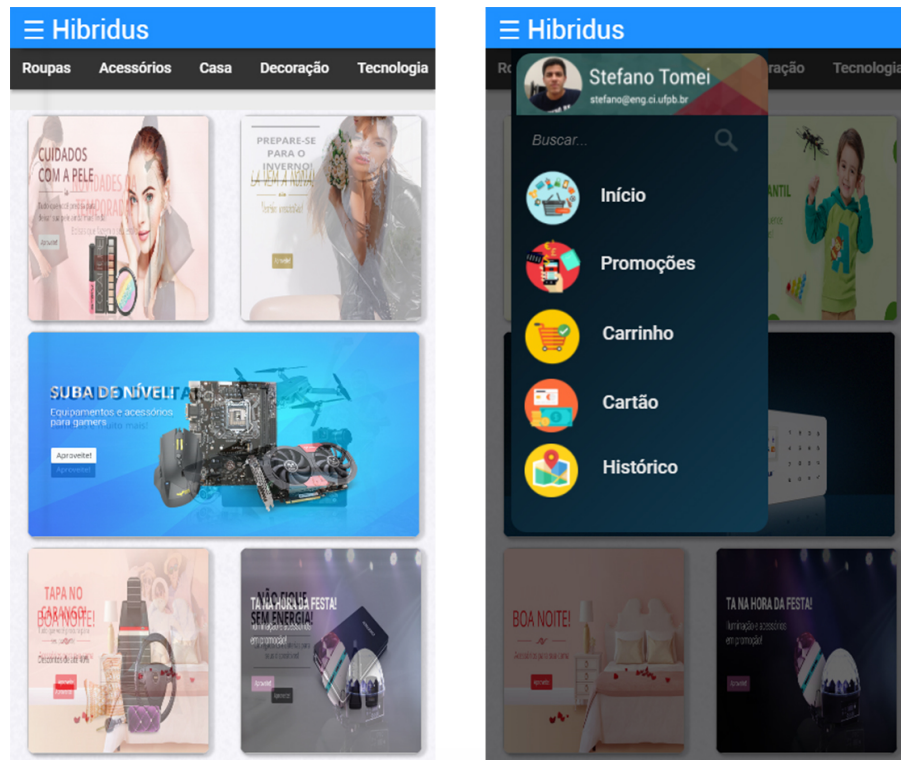
Fonte: Autor

Figura 17: Resultado iPhone 6



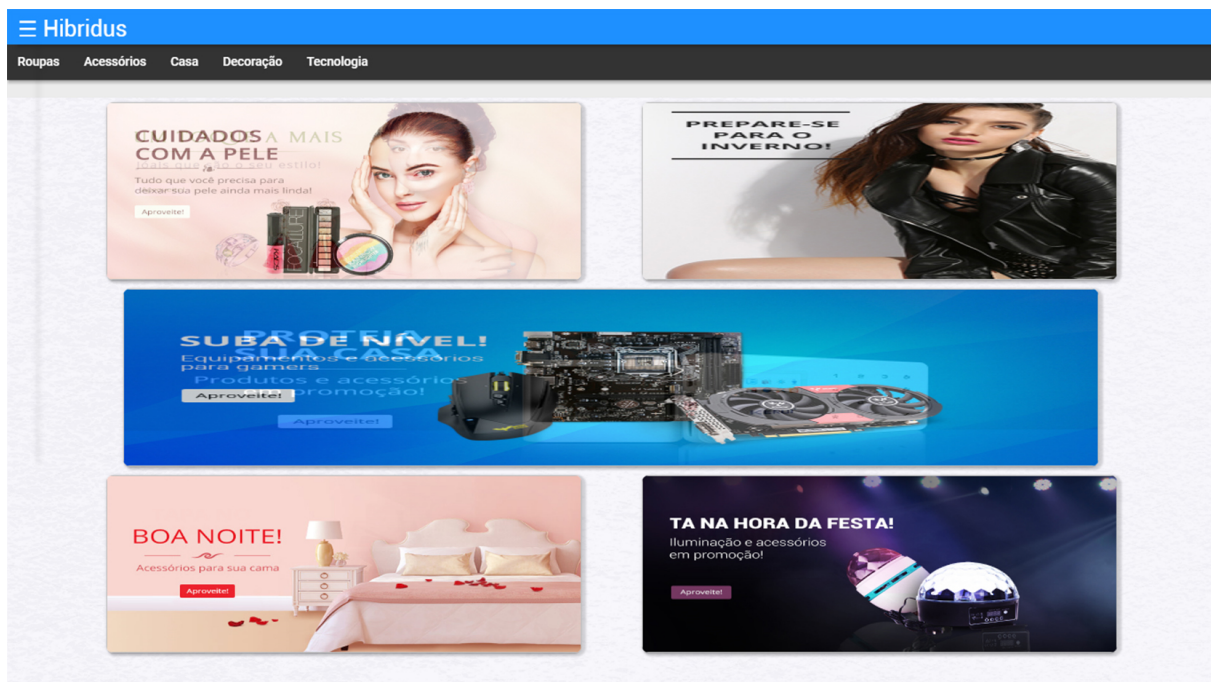
Fonte: Autor

Figura 18: Resultado iPhone 6 Plus



Fonte: Autor

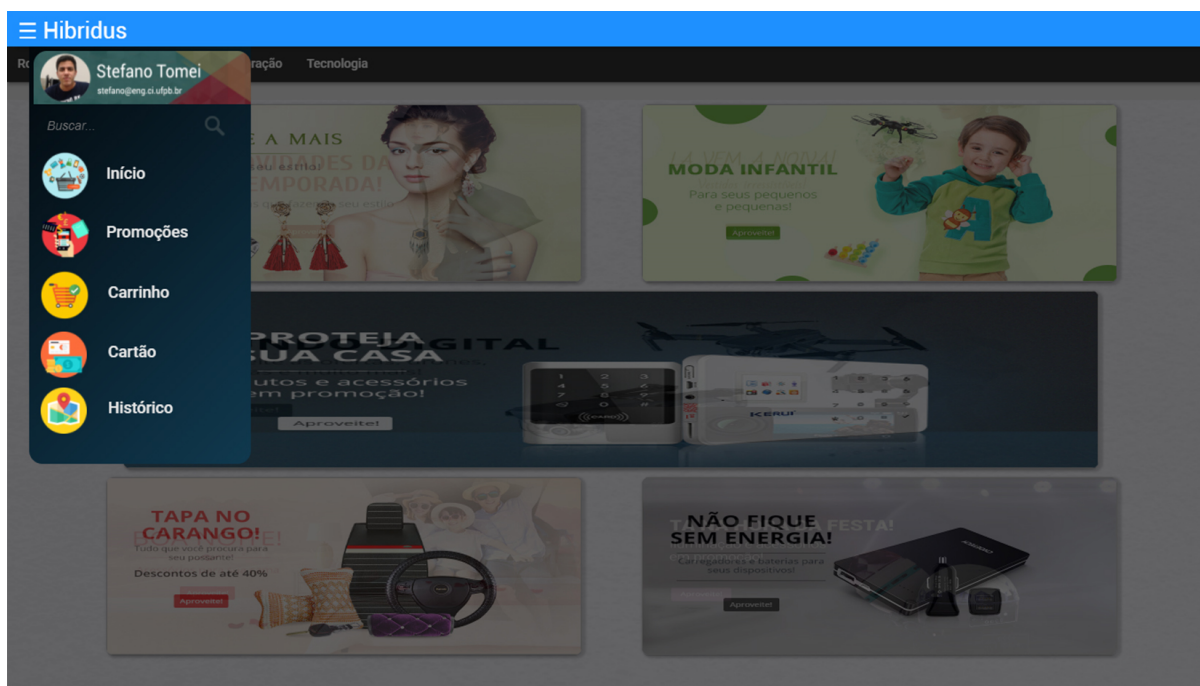
Figura 19: Resultado Desktop 1



Fonte: Autor



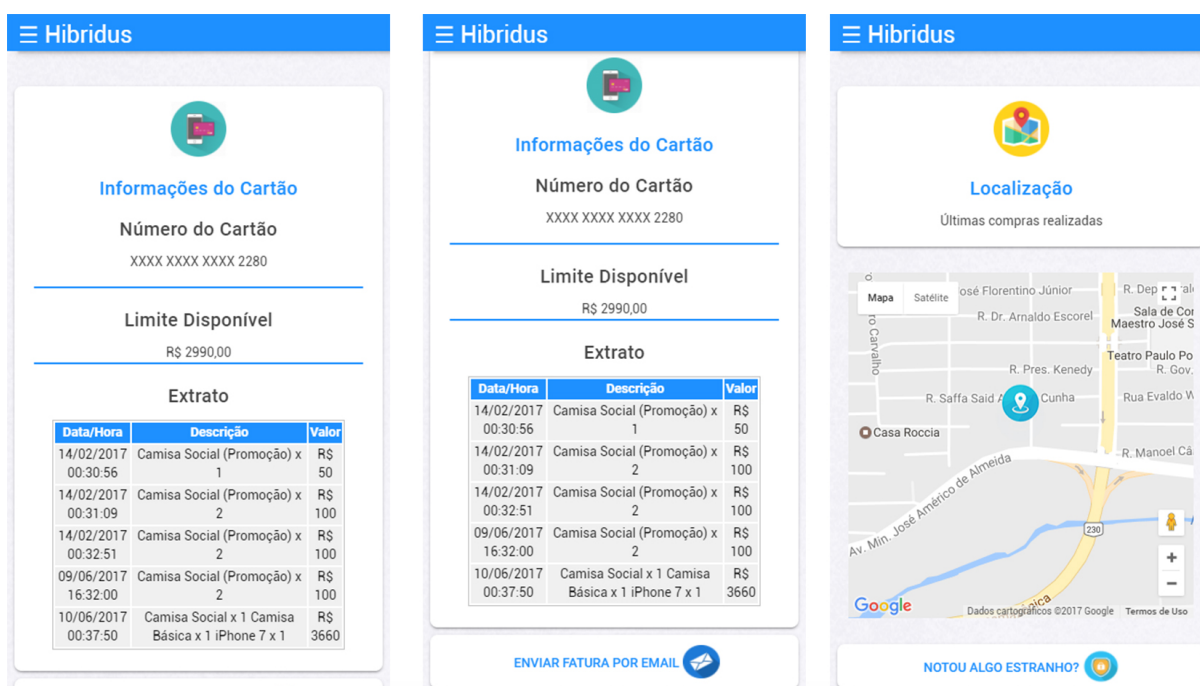
Figura 20: Resultado Desktop 2



Fonte: Autor

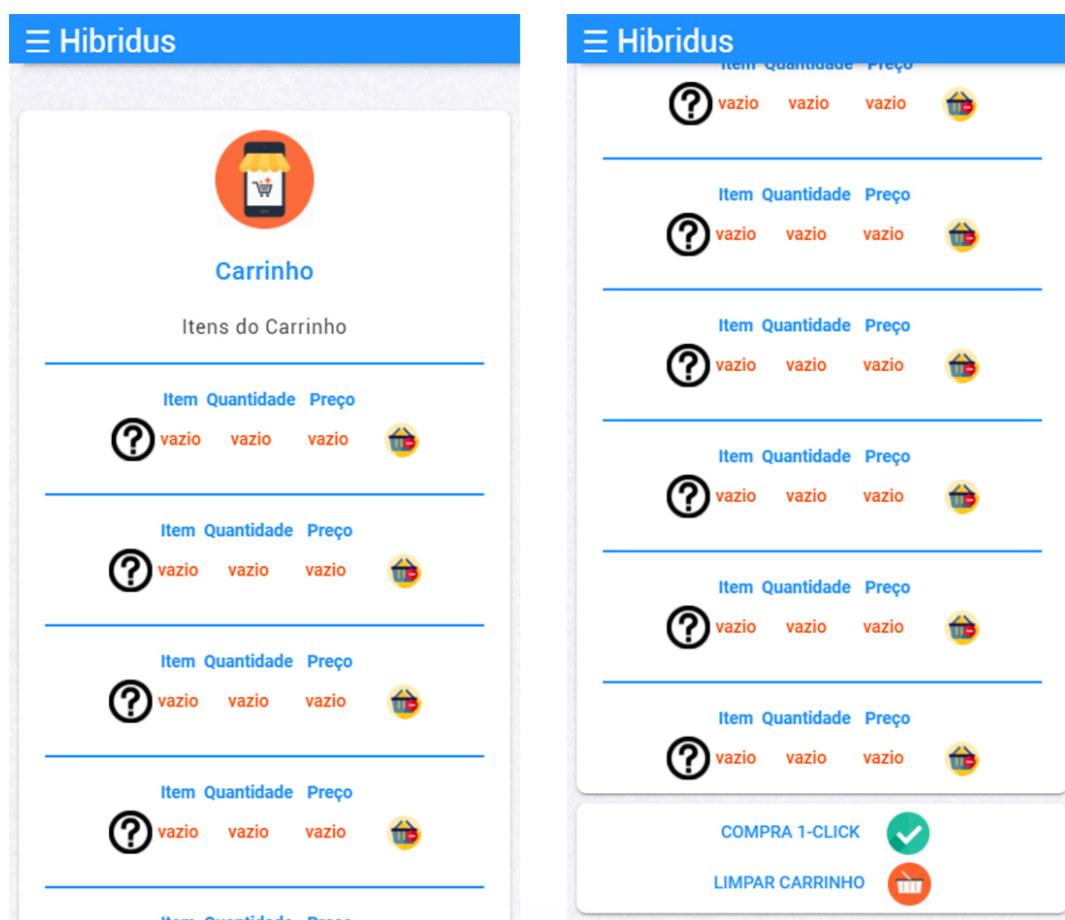
Do mesmo modo os resultados se aplicam as telas de carrinho, cartão e localização. Utilizando de uma resolução intermediária que é a do iPhone 6 Plus, serão exibidos os resultados para essas três páginas a seguir:

Figura 21: Resultado Informações do cartão e Localização no iPhone 6 Plus



Fonte: Autor

Figura 22: Resultado Carrinho no iPhone 6 Plus



Fonte: Autor

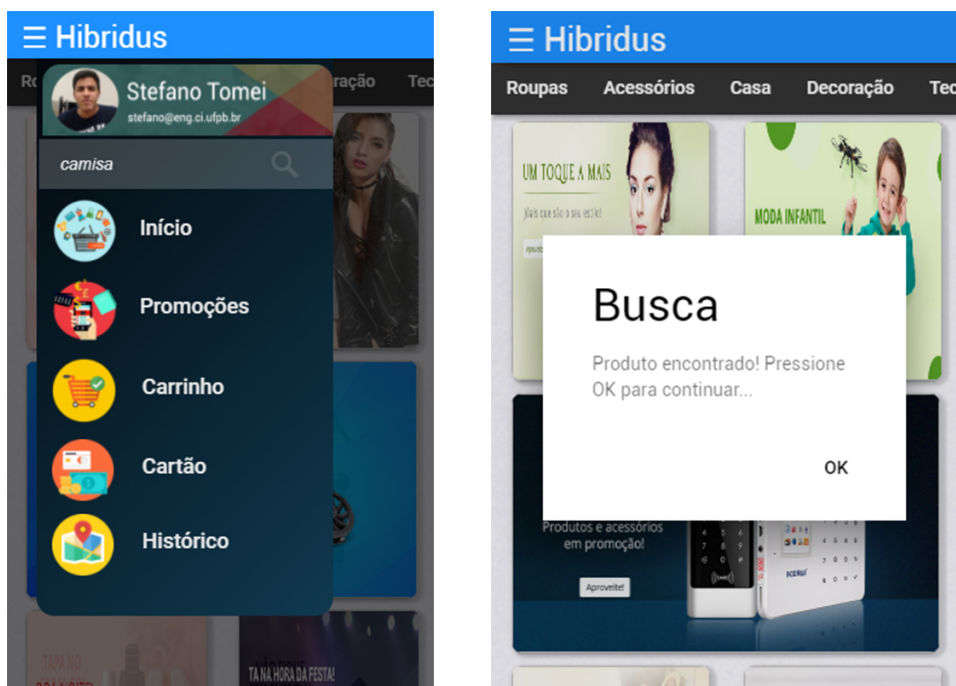
As funcionalidades foram todas feitas utilizando JavaScript, desde as validações de compra até o armazenamento da posição atual do dispositivo. Todo o armazenamento de informações é feito através de Web Storage, recurso disponível no HTML 5 que substitui parcialmente a necessidade de um banco de dados para demonstração de funcionamento do sistema.

Todos os testes foram feitos utilizando o gulp para criar um socket e a partir disso serem testadas as compras e o comportamento de cada página. Os dispositivos físicos utilizados para experimento foram: Moto G 1ª e 3ª geração, Tablet Samsung Galaxy Tab, iPhone 4S, iPhone 6S e Samsung Galaxy S7.

Para validação de experimentos baseado em dispositivos físicos, foi utilizado o Samsung Galaxy S7 cuja resolução é de 1440 x 2560 pixel e ajustado automaticamente, permitindo efetuar compras e exibindo todas as funcionalidades do PWA. Serão exibidas as etapas desde a pesquisa pelo produto até a verificação de validação do procedimento.

Inicialmente, é feita a pesquisa pelo produto desejado e o produto é encontrado (Figura 23):

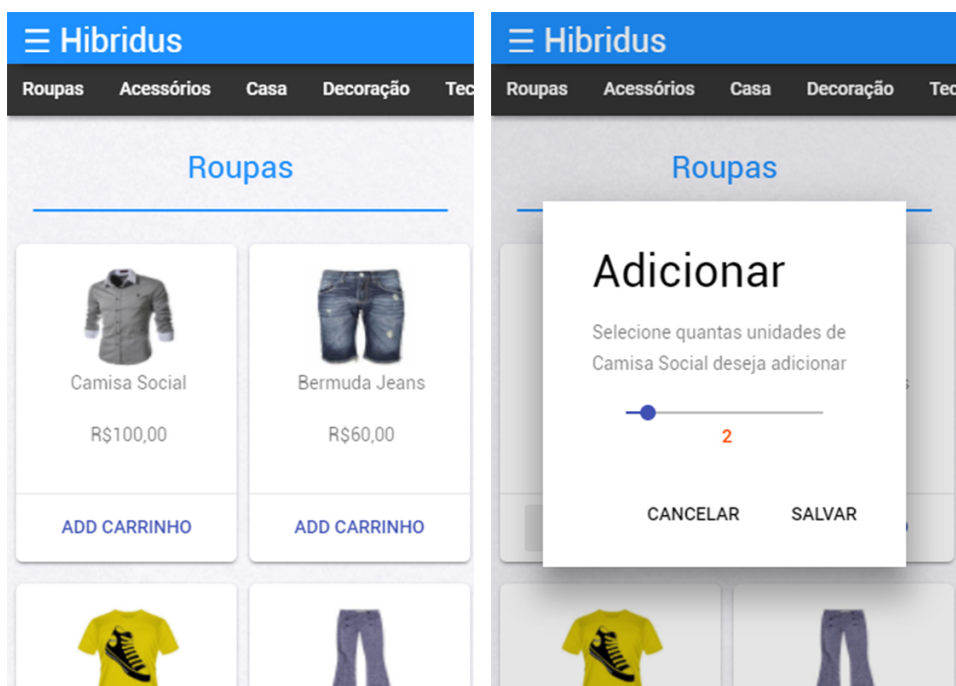
Figura 23: Resultado de busca por produto "camisa"



Fonte: Autor

Em seguida, é selecionado o item e quantas unidades deseja comprar (**Figura 24**):

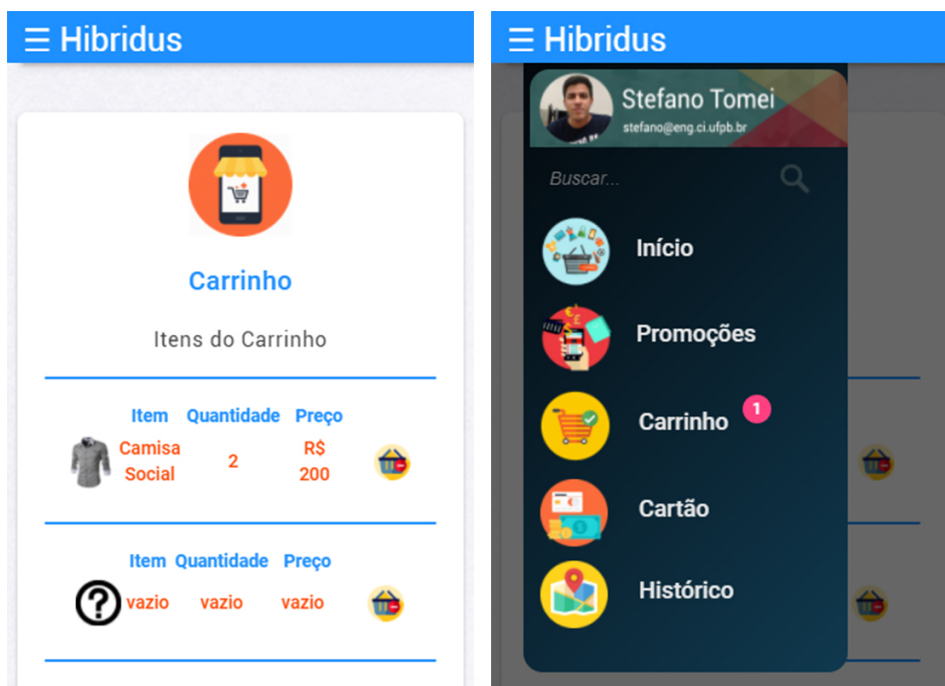
Figura 24: Produto adicionado ao carrinho



Fonte: Autor

O usuário é redirecionado para o carrinho e número de itens atualizado (**Figura 25**):

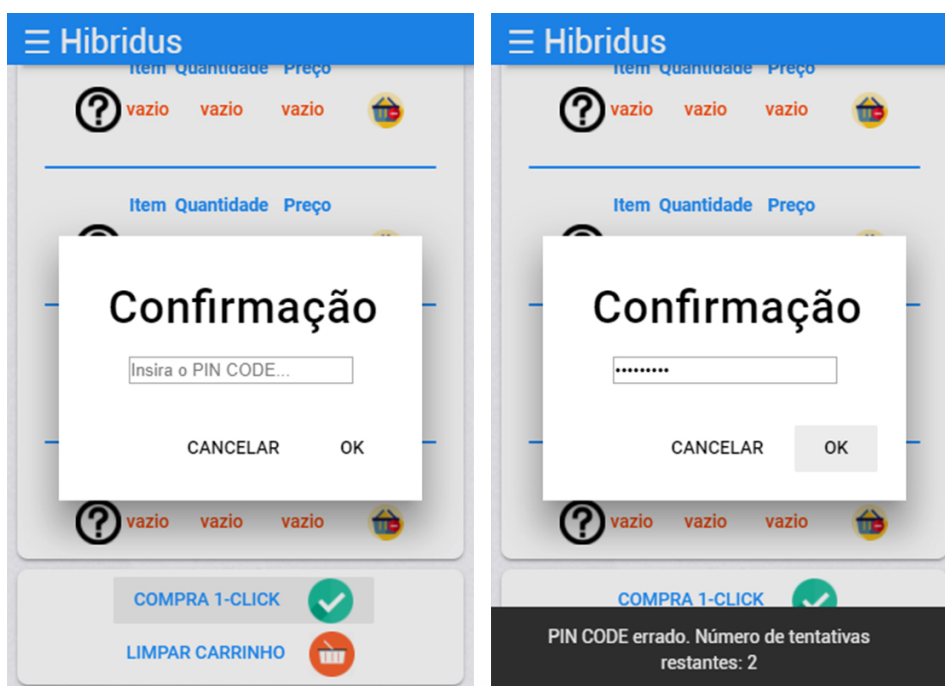
Figura 25: Item adicionado ao carrinho



Fonte: Autor

Durante a validação da compra, foi inserido um código PIN incorreto e exibida a respectiva mensagem do erro (**Figura 26**):

Figura 26: Validação de compra

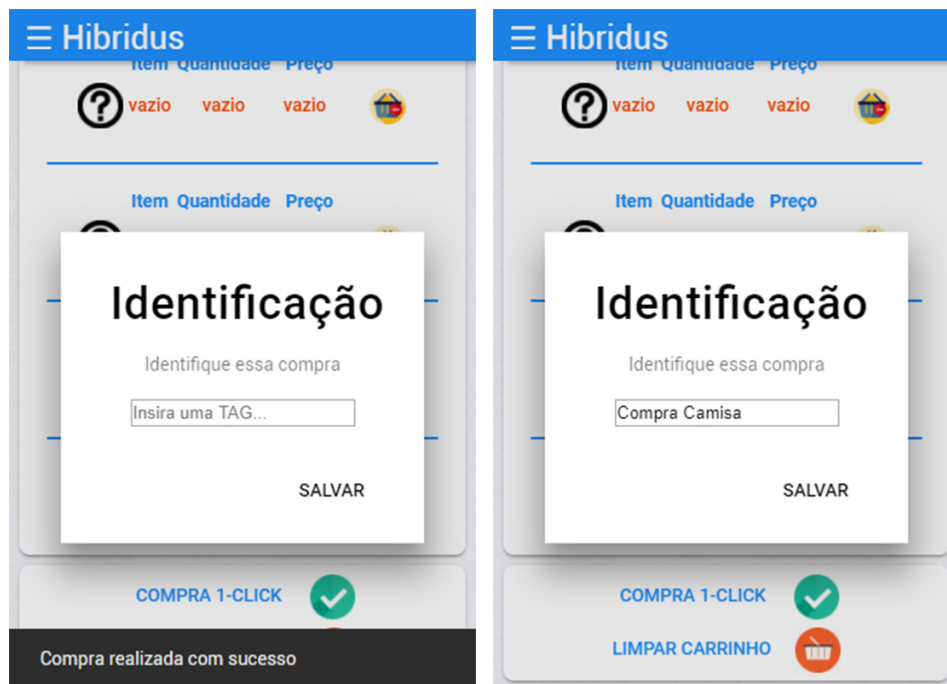


Fonte: Autor

Inserido corretamente o PIN, é solicitada uma Tag para identificação da compra

(Figura 27):

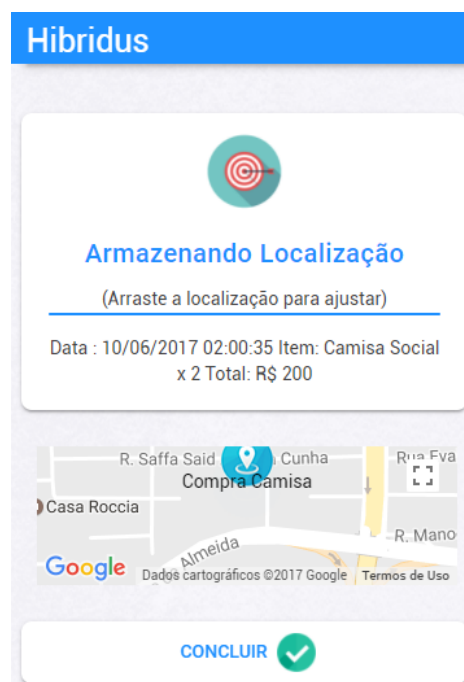
Figura 27: Compra confirmada



Fonte: Autor

Após definir a Tag, são conferidos os dados da compra. A partir disso, será gerado um marcador com essa Tag (Figura 28):

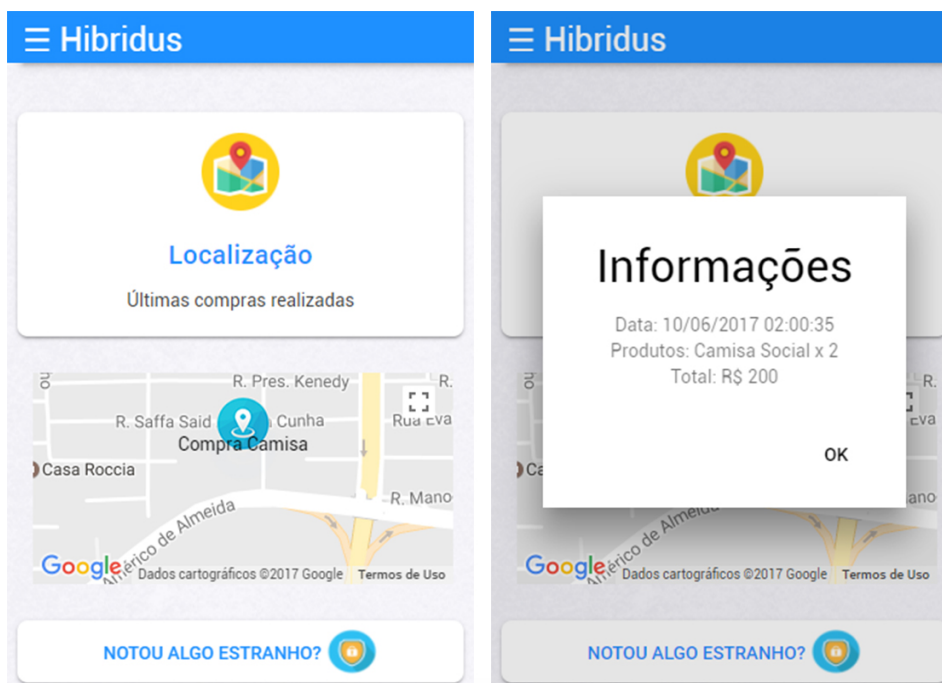
Figura 28: Verificação de informações da compra



Fonte: Autor

Após concluir a compra, é possível então observar no mapa onde foi realizada e ao clicar sob a localização, são exibidas as informações da compra (**Figura 29**):

**Figura 29: Informações no local da compra**



Fonte: Autor

O valor foi descontado do limite e a fatura enviada (**Figura 30**):

**Figura 30: Informações do cartão**



Fonte: Autor



## 5 CONCLUSÕES E TRABALHOS FUTUROS

Baseado nos resultados apresentados no capítulo anterior, conclui-se que o PWA resolveu o problema de responsividade e operacionalidade em dispositivos computacionais, tendo em vista que houve a correta adaptação de layout e total funcionamento da aplicação.

A solução foi testada em vários dispositivos de diferentes resoluções e apresentou o comportamento esperado, sem apresentar problemas quanto a execução ou adaptação visual. Com isso, é um fato que a modelagem realizada através de HTML e CSS foi fundamental para que fosse alcançado o objetivo de solucionar o problema de responsividade e operacionalidade.

Para trabalhos futuros, a fim de melhorar esteticamente, utilizar de outras ferramentas como templates apropriados para mercado virtual. É interessante também obter informações derivadas de banco de dados, isso se dá através de inserção de um back-end onde haja consumo de uma API voltada para esse tipo de aplicação. Comumente nesse tipo de mercado, o back-end entra na parte de listar os produtos e armazenar as informações após a compra ser realizada, o que de fato levaria até a empresa o que foi comprado e subtraído de seu estoque o produto adquirido pelo usuário.

## REFERÊNCIAS

- [1] PORTO, Fabiano. Aplicativos Mobile: Definições, história e previsões. São Paulo: Tec Triade Brasil, 2012. Disponível em: <<http://tectriadebrasil.com.br/blog/mercado-de-midias-sociais-blog/aplicativos-mobile-definicoes-historia-e-previsoes/>>. Acesso em: 5 de Junho de 2017.
- [2] BENFICA, Alex. O que é app?. Paraná: Telefones Celulares, 2009. Disponível em: <<https://www.telefonescelulares.com.br/o-que-e-app/>>. Acesso em: 7 de Junho de 2017.
- [3] BOZZA, Cláudia. Saiba o que é um navegador e um sistema operacional. São Paulo: Techtudo, 2013. Disponível em: <<http://www.techtudo.com.br/dicas-e-tutoriais/noticia/2011/08/saiba-o-que-e-um-navegador-e-um-sistema-operacional.html>>. Acesso em: 5 de Junho de 2017.
- [4] ARRIGONI, Ricardo. Responsive Design: dicas para tornar seu site acessível em qualquer resolução. São Paulo: DevMedia, 2016. Disponível em: <<http://www.devmedia.com.br/responsive-design-dicas-para-tornar-seu-site-acessivel-em-qualquer-resolucao/28316>>. Acesso em: 6 de Fevereiro de 2017.
- [5] MOREIRA, Eduardo. Os navegadores web mais usados em dispositivos móveis. São Paulo: TargetHD, 2015. Disponível em: <<http://targethd.net/os-navegadores-web-mais-usados-em-dispositivos-moveis-em-abril-de-2015/>>. Acesso em: 7 de Fevereiro de 2017.
- [6] LEPAGE, Pete. Seu primeiro Progressive Web App. USA: Google, 2017. Disponível em: <<https://developers.google.com/web/fundamentals/getting-started/codelabs/your-first-pwapp/>>. Acesso em: 14 de Novembro de 2016.
- [7] CASTELLI, Ian. Material Design: um olhar aprofundado sobre o novo estilo visual da Google . Brasil: Google, 2014. Disponível em: <<https://www.tecmundo.com.br/google/58278-material-design-olhar-aprofundado-novo-estilo-visual-google.htm>>. Acesso em: 16 de Novembro de 2016.
- [8] HAMANN, Renan. iOS, Android e Windows Phone: números dos gigantes comparados. Brasil: Tecmundo, 2014. Disponível em: <<https://www.tecmundo.com.br/sistema-operacional/60596-ios-android-windows-phone-numeros-gigantes-comparados-infografico.htm>>. Acesso em: 8 de Junho de 2017.
- [9] EIS, Diego. Introdução sobre Media Queries. Brasil: Tecmundo, 2009. Disponível em: <<https://tableless.com.br/introducao-sobre-media-queries/>>. Acesso em: 15 de Novembro de 2016.



- [10] FIRDAUS, Thoriq. Get up and Running with Google Web Starter Kit. N/A: Tutsplus, 2014. Disponível em: <<https://webdesign.tutsplus.com/tutorials/get-up-and-running-with-google-web-starter-kit-cms-21495>>. Acesso em: 15 de Novembro de 2016.
- [11] ALBUQUERQUE, Rodney. Como criar páginas responsivas para vários dispositivos. Brasil: Linhadecódigo, 2016. Disponível em: <<http://www.linhadecodigo.com.br/artigo/3628/como-criar-paginas-responsivas-para-varios-dispositivos.aspx>>. Acesso em: 4 de Dezembro de 2016.
- [12] CAPISTRANO, Daniela. Mas afinal, o que é um App?. Brasil: Fábrica de Aplicativos, 2012. Disponível em: <<http://fabricadeaplicativos.com.br/fabrica/mas-afinal-o-que-e-um-app/>>. Acesso em: 8 de Junho de 2017.
- [13] FERREIRA, Davi. React: JavaScript reativo, 2013. Disponível em: <<https://tableless.com.br/react-javascript-reativo/>>. Acesso em: 8 de Junho de 2017.

## APÊNDICE A

### 5.1 Funcionalidades da barra de navegação

Para a barra de navegação associada ao menu, temos a opção de abrir e fechar. Toda essa funcionalidade é implementada no arquivo *navigation.js*. Ao abrir o menu, é inserida uma sombra nos demais elementos da tela e ao fechar essa sombra é eliminada e o menu recolhido.

```
1  function openNav() {
2      var cont = parseInt(window.sessionStorage.getItem("cont"));
3      if (cont == 0){
4          document.getElementById("myNavigation").style.width = "0";
5          document.getElementById("myNavigation").style.boxShadow = '10px
6              2px 9px 1px rgba(0, 0, 0, 0.15)';
7          window.sessionStorage.setItem("cont",1);
8      }
9      else{
10         document.getElementById("myNavigation").style.width = "70%";
11         document.getElementById("myNavigation").style.boxShadow = '400px
12             800px 0px 800px rgba(0, 0, 0, 0.6)';
13         window.sessionStorage.setItem("cont",0);
14     }
15 }
16
17 function closeNav() {
18     document.getElementById("myNavigation").style.width = "0";
19     document.getElementById("myNavigation").style.boxShadow = '0px 0
20         px 0px 0px rgba(0, 0, 0, 0)';
21     window.sessionStorage.setItem("cont",1);
22 }
```

---

Para cada ícone temos uma referência associada para que, ao usuário clicar no mesmo, seja redirecionado para a respectiva página.

```
1  function openHome() {
2      closeNav();
3      window.location.href = "index.html";
4  }
5
6  function openPromo() {
7      closeNav();
8      window.location.href = "promotion.html";
9  }
```

```

10
11 function openCar() {
12     closeNav();
13     window.location.href = "shopping-cart.html";
14 }
15
16 function openCard() {
17     closeNav();
18     window.location.href = "card.html";
19 }
20
21 function openHist() {
22     window.location.href = "localization_load.html";
23     closeNav();
24 }

```

---

Para a funcionalidade de busca, utiliza-se a função *openBusca(inter)* onde baseado na palavra chave inserida, o usuário será redirecionado para a página da categoria do produto.

```

1 function openBusca(inter) {
2     closeNav();
3     var txtRead = document.getElementById("txtBusca").value;
4     var txtBusca = txtRead.substring(0,1).toUpperCase() +
        txtRead.substring(1);
5
6     var dialog_busca = window.document.getElementById('busca-dialog'
        );
7     var busca_content = window.document.getElementById('
        busca-content');
8
9     if (! dialog_busca.showModal) {
10        dialogPolyfill.registerDialog(dialog_busca);
11    }
12    dialog_busca.showModal();
13
14    if ( txtBusca == "Roupa" || txtBusca == "Roupas" || txtBusca == "
        Camisa" || txtBusca == "Bermuda" || txtBusca == "Calca" ||
        txtBusca == "Sunga de banho") {
15
16        busca_content.textContent = "Produto encontrado! Pressione OK
        para continuar...";
17        document.querySelector('.close').addEventListener('click',
        function() {
18            window.location.href = "clothing.html";});
19    }

```

```

20     else if ( txtBusca == "Tecnologia" || txtBusca == "Digital" ||
        txtBusca == "Tablet" || txtBusca == "iPhone" || txtBusca == "
        Samsung" || txtBusca == "Multimidia" || txtBusca == "Som" ||
        txtBusca == "Projektor") {
21         busca_content.textContent = "Produto encontrado! Pressione OK
            para continuar...";
22         document.querySelector('.close').addEventListener('click'
            ', function() {
23             window.location.href = "technology.html";});
24     }
25     //Caso NAO NCONTRE
26     else {
27         busca_content.textContent = "Produto nao encontrado";
28         document.querySelector('.close').addEventListener('click'
            ', function() {
29             dialog_busca.close();});
30         document.getElementById("txtBusca").value = "";
31     }
32 }

```

---

## 5.2 Implementação do processo de compra

Para melhor compreensão do processo de compra, os códigos a seguir exibem como foram feitas todas as funções de validação do procedimento.

Para adicionar o item ao carrinho, é necessário ocupar um slot disponível no armazenamento selecionado, que foi o Session Storage, para que ao fim da navegação, os dados sejam excluídos. Para isso, utiliza-se a função *getItemValues(name, src, value)* contendo o nome do produto, imagem e valor total (relação valor x quantidade de itens).

Caso a operação seja cancelada, é então exibida a mensagem de cancelamento através de um toast. Também há o tratamento para o carrinho cheio.

```

1     function getItemValues(name, src, value) {
2         var item_name = window.sessionStorage.getItem("name_1");
3         if (item_name != null) {
4             getItemValues_2(name, src, value);
5         }
6         else {
7
8             var cart_dialog = window.document.getElementById('cart-dialog');
9             var cart_dialog_content = document.getElementById('cart-content'
                ');
10
11             var slider_value = document.getElementById('slider-value');

```

```

12     var txt_input = document.getElementById('txt-input');
13
14     // Atualizar o numero abaixo do slider
15     slider_value.addEventListener('input', function () {
16         txt_input.innerText = slider_value.value;
17     }, false);
18
19     // Cancelamento de adicionar
20     if (! cart_dialog.showModal) {
21         dialogPolyfill.registerDialog(cart_dialog);
22     }
23     cart_dialog.showModal();
24
25     cart_dialog.querySelector('.close').addEventListener('click',
26         function(){
27         msgToast('cancelada');
28         cart_dialog.close();
29     });
30
31     cart_dialog_content.querySelector('p').innerText = 'Selecione
32         quantas unidades de '+ name +' deseja adicionar ';
33
34     cart_dialog.querySelector('.confirm-cart').addEventListener('
35         click',function(){
36
37         qtde = slider_value.value;
38         // Armazenamento dos dados
39         window.sessionStorage.setItem("name_1", name);
40         window.sessionStorage.setItem("src_1", src);
41         window.sessionStorage.setItem("value_1", value * qtde);
42         window.sessionStorage.setItem("quantity_1", qtde);
43         window.location.href = "shopping-cart.html";
44     });
45 }
46
47 // Avisos para carrinho cheio ou cancelamento
48 function msgToast(status) {
49
50     if(status == 'cancelada'){
51
52         var snackbarContainer = document.querySelector('#toast-clear')
53         ;
54         var showToastButton = document.querySelector('#show-toast');
55         var data = {message: 'Cancelado'}; //# ' + ++counter
56         snackbarContainer.MaterialSnackbar.showSnackbar(data);
57     }

```

```

55     if(status == 'lotado'){
56
57         var snackbarContainer = document.querySelector('#toast-clear')
58         ;
59         var showToastButton = document.querySelector('#show-toast');
60         var data = {message: 'Carrinho LOTADO! Confira seu carrinho'};
61         snackbarContainer.MaterialSnackbar.showSnackbar(data);
62     }
63 }

```

---

Para a tela de carrinho exibir as informações armazenadas, utiliza-se da função *loadCart()*.

```

1  function loadCart() {
2
3  var div_nome1 = document.getElementById("nome_item1");
4  div_nome1.textContent = window.sessionStorage.getItem("name_1");
5  // Verifica se ta vazio
6  if (div_nome1.textContent != "") {
7
8      var load = new Image();
9      load.src = window.sessionStorage.getItem("src_1");
10     document.getElementById("icone_item1").innerHTML = "<img src=\"\"
11         + load.src + \"\" width=35 />";
12
13     var div_valor = document.getElementById("valor_item1");
14     div_valor.textContent = "R$ " + window.sessionStorage.getItem("
15         value_1");
16
17     var div_qtde = document.getElementById("qtde_item1");
18     div_qtde.textContent = window.sessionStorage.getItem("quantity_1
19         ");
20 } // Caso seja vazio, retorna vazio
21 else {
22     document.getElementById("nome_item1").textContent = "vazio";
23     document.getElementById("icone_item1").innerHTML = "<img src=\"'
24         images/cart/empty.png' width=35 />";
25     document.getElementById("valor_item1").textContent = "vazio";
26     document.getElementById("qtde_item1").textContent = "vazio";
27 }
28 }

```

---

Para remover itens do carrinho, utiliza-se a função *removeItem(item\_id)* que recebe como parâmetro em que botão foi clicado para remover o respectivo item do carrinho.

```

1  function removeItem(item_id) {
2
3  if (item_id == "item_1") {
4      if(window.sessionStorage.getItem("name_1") == null){
5          clearToast('clean');
6      } else {
7          var dialog_1 = document.getElementById('remover-1');
8          var showDialogButton_1 = document.querySelector('#show-dialog-1'
9              );
10
11         if (! dialog_1.showModal) {
12             dialogPolyfill.registerDialog(dialog_1);
13         }
14         dialog_1.showModal();
15         dialog_1.querySelector('.close').addEventListener('click',
16             function(){
17                 dialog_1.close();
18             });
19
20         window.document.getElementById("remover-item-1").textContent =
21             window.sessionStorage.getItem("name_1");
22         document.querySelector('.confirm-1').addEventListener('click',
23             function() {
24                 window.sessionStorage.removeItem("name_1");
25                 window.sessionStorage.removeItem("src_1");
26                 window.sessionStorage.removeItem("value_1");
27                 window.sessionStorage.removeItem("quantity_1");
28                 window.location.href = "shopping-cart.html";});
29     }
30 }

```

---

Ao selecionar a opção comprar, são lidas as informações dos itens do carrinho. Os campos que contém valores são considerados para que, a partir daí, seja iniciado o processo de conclusão da compra. São permitidos no máximo seis itens no carrinho:

```

1  function buyInfos(value_1, value_2, value_3, value_4, value_5,
2      value_6, total, tag) {
3
4      var name_qtde;
5      var date = new Date();
6      var data = date.toLocaleDateString() + " " +
7          date.toLocaleTimeString();
8      var geolocation = window.navigator.geolocation;
9
10     if (value_1 != 0) {

```

```

9         name_qtde = window.sessionStorage.getItem("name_1") +
10             " x " + window.sessionStorage.getItem("quantity_1");
11         var result = (window.localStorage.getItem("limite_atual"
12             ) - value_1);
13     }
14     // Local para armazenamento
15     if (window.localStorage.getItem("extrato_infos_1") != null) {
16         posicao2();
17     } else {
18         window.localStorage.setItem("extrato_infos_1", name_qtde);
19         window.localStorage.setItem("extrato_total_1", total);
20         window.localStorage.setItem("extrato_date_1", data);
21         window.localStorage.setItem("limite_atual", result);
22         window.localStorage.setItem("tag_1", tag);
23     }

```

---

É então iniciado o processo de confirmação de compra. O PIN inserido é verificado para que seja de acordo com o pré-definido. Caso não seja, é retornada a mensagem de erro e o número de tentativas restantes. Caso esse número esgote ou o limite esteja estourado, a compra é cancelada e o cartão bloqueado.

O limite disponível para compra e status do cartão é verificado através dos valores definidos para o cartão em questão. Esses valores são pré-armazenados e verificados antes da conclusão da compra.

```

1     function buyClick() {
2         var snackbarContainer = document.getElementById('toast-clear')
3         ;
4         var showToastButton = document.getElementById('show-toast');
5         // Recebendo os valores
6         var value_1 = window.sessionStorage.getItem("value_1");
7         var value_2 = window.sessionStorage.getItem("value_2");
8         var value_3 = window.sessionStorage.getItem("value_3");
9         var value_4 = window.sessionStorage.getItem("value_4");
10        var value_5 = window.sessionStorage.getItem("value_5");
11        var value_6 = window.sessionStorage.getItem("value_6");
12        // Verificando se sao nulos
13        if (value_1 == null) {
14            value_1 = 0;
15        }
16        if (value_2 == null) {
17            value_2 = 0;
18        }
19        if (value_3 == null) {
20            value_3 = 0;
21        }
22    }

```



```

21     if (value_4 == null) {
22         value_4 = 0;
23     }
24     if (value_5 == null) {
25         value_5 = 0;
26     }
27     if (value_6 == null) {
28         value_6 = 0;
29     }
30     // Somando os valores
31     var total = parseInt(value_1) + parseInt(value_2) + parseInt(
32         value_3)
33         + parseInt(value_4) + parseInt(value_5) + parseInt(value_6
34             );
35     // Validando a compra
36     var PIN_CODE = "A1B2C3D4";
37     var code;
38     var tag;
39     var limite;
40     var tentativas;
41
42     if (total == 0){
43         clearToast('vazio');
44     } else {
45         tentativas = window.localStorage.getItem("tentativas");
46         if (tentativas != 0 && tentativas > 0) {
47             var buy_validate = window.document.getElementById(
48                 'buy-content');
49             var dialog_buy = document.getElementById('validate
50                 ');
51
52             if (! dialog_buy.showModal) {
53                 dialogPolyfill.registerDialog(dialog_buy);
54             }
55             dialog_buy.showModal();
56
57     dialog_buy.querySelector('.close').addEventListener('click',
58         function(){
59             clearToast('cancelada');
60             dialog_buy.close();
61         });
62
63     buy_validate.innerHTML = '<input type="password"
64         id="txtPINCODE" placeholder="Insira o PIN
65         CODE..." />';
66
67     // Validando a compra

```

```

61         document.querySelector('.confirm-buy').
           addEventListener('click', function() {
62             // Concluir compra capturando o valor do input
63             code = (window.document.getElementById('
               txtPINCODE').value).toUpperCase();
64             if (PIN_CODE == code) {
65                 document.getElementById('closebtn').
                   style.display = 'none';
66                 limite = window.localStorage.getItem("
                   limite_atual") - total;
67
68                 if (limite >= 0){
69                     // Caso a compra seja um sucesso
70                     clearToast('sucesso');
71
72                     var pin_txt =
                       document.getElementById('
                         txtPINCODE');
73
74                     pin_txt.parentNode.removeChild(
                       pin_txt);
75
76                     dialog_buy.querySelector('h5').
                       innerText = 'Identificacao';
77                     buy_validate.innerHTML = '
                       Identifique essa compra'+
78                     '<p></p><input type="text" id="
                       txtTAG" placeholder="Insira uma
                       TAG..." />';
79
80                     document.getElementById('closebtn'
                       ).style.display = 'none';
81                     document.getElementById('
                       confirmbtn').style.display = '
                       none';
82                     document.getElementById('savebtn')
                       .style.display = 'inline';
83
84                     // Segue o fluxo
85                     document.querySelector('.save').
                       addEventListener('click',
                       function() {
86                         window.localStorage.setItem("
                           tentativas" , 3);
87                         tag =
                           window.document.getElementById
                             ('txtTAG').value;

```

```

88         buyInfos(value_1, value_2,
89                 value_3, value_4, value_5,
90                 value_6, total, tag);
91     });
92     } else{
93         // Caso o limite tenha estourado
94         clearToast('limite');
95         dialog_buy.close();
96     }
97 } else {
98     // Caso o PIN_CODE seja errado
99     tentativas--;
100    clearToast('pincode');
101    window.localStorage.setItem("tentativas" ,
102                                tentativas);
103    tentativas = window.localStorage.getItem("
104                                tentativas");
105    if(tentativas == 0){
106        window.localStorage.setItem("
107                                tentativas" , 0);
108        dialog_buy.close();
109        clearToast('esgotada');
110        clearCache();
111    }
112 }
113 });
114 } else {
115     // Esgotadas as tentativas
116     clearToast('esgotada');
117     clearCache();
118 }
119 }
120 }

```

---

Após a conclusão da compra, é limpo o cache do Session Storage com a função *clearCache(item\_id)* ou para remover item do carrinho. Em seguida, é feito o armazenamento da posição atual do usuário.

```

1 function clearCache(item_id) {
2
3     var value_1 = window.sessionStorage.getItem("value_1");
4     var value_2 = window.sessionStorage.getItem("value_2");
5     var value_3 = window.sessionStorage.getItem("value_3");
6     var value_4 = window.sessionStorage.getItem("value_4");

```

```

7  var value_5 = window.sessionStorage.getItem("value_5");
8  var value_6 = window.sessionStorage.getItem("value_6");
9
10 if (item_id == "limpar") {
11
12     if((value_1 == null) || (value_2 == null) || (value_3 == null)
13         ||
14         (value_4 == null) || (value_5 == null) || (value_6 == null)){
15         clearToast('vazio');
16     } else {
17         var clear_validate = window.document.getElementById('
18             clear-content');
19         var dialog_clear = document.getElementById('clear');
20
21         if (! dialog_clear.showModal) {
22             dialogPolyfill.registerDialog(dialog_clear);
23         }
24         dialog_clear.showModal();
25
26         dialog_clear.querySelector('.close').addEventListener('click',
27             function(){
28                 clearToast('cancelado');
29                 dialog_clear.close();
30             });
31         dialog_clear.querySelector('.confirm-clear').addEventListener(
32             'click',function(){
33                 clearToast('vazio');
34                 window.sessionStorage.clear();
35                 window.location.href = "shopping-cart.html";
36                 dialog_clear.close();
37             });
38     }
39 }
40 if (item_id == "comprar") {
41
42     window.sessionStorage.clear();
43     // Salvar o local atual
44     window.location.href = "localization_save.html";
45 }
46 }

```

---

As mensagens de status podem ser chamadas através da função seguir de nome *clearToast(status)*:

```

1  function clearToast(status) {
2

```

```

3     switch (status){
4
5         case 'vazio':
6             var snackbarContainer = document.querySelector('#toast-clear
7                 ');
8             var showToastButton = document.querySelector('#show-toast');
9             var data = {message: 'Carrinho vazio'}; //# ' + ++counter
10            snackbarContainer.MaterialSnackbar.showSnackbar(data);
11            break;
12
13        case 'limite':
14            var snackbarContainer = document.querySelector('#toast-clear
15                ');
16            var showToastButton = document.querySelector('#show-toast');
17            var data = {message: 'Limite estourado! Confira seu limite e
18                tente novamente'};
19            snackbarContainer.MaterialSnackbar.showSnackbar(data);
20            break;
21
22        case 'sucesso':
23            var snackbarContainer = document.querySelector('#toast-clear
24                ');
25            var showToastButton = document.querySelector('#show-toast');
26            var data = {message: 'Compra realizada com sucesso'};
27            snackbarContainer.MaterialSnackbar.showSnackbar(data);
28            break;
29
30        case 'cancelada':
31            var snackbarContainer = document.querySelector('#toast-clear
32                ');
33            var showToastButton = document.querySelector('#show-toast');
34            var data = {message: 'Compra cancelada'};
35            snackbarContainer.MaterialSnackbar.showSnackbar(data);
36            break;
37
38        case 'cancelado':
39            var snackbarContainer = document.querySelector('#toast-clear
40                ');
41            var showToastButton = document.querySelector('#show-toast');
42            var data = {message: 'Limpeza cancelada'};
43            snackbarContainer.MaterialSnackbar.showSnackbar(data);
44            break;
45
46        case 'pincode':
47            var snackbarContainer = document.querySelector('#toast-clear
48                ');
49            var showToastButton = document.querySelector('#show-toast');
50            var data = {message: 'Pincode incorreto'};
51            snackbarContainer.MaterialSnackbar.showSnackbar(data);
52            break;
53    }
54 }

```

```

43     var tentativas = window.localStorage.getItem("tentativas")
        -1;
44     if(tentativas == 0){break;}
45     var data = {message: 'PIN CODE errado. Numero de tentativas
        restantes: ' + tentativas};
46     snackbarContainer.MaterialSnackbar.showSnackbar(data);
47     break;
48
49     case 'esgotada':
50     var snackbarContainer = document.querySelector('#toast-clear
        ');
51     var showToastButton = document.querySelector('#show-toast');
52     var data = {message: 'Numero de tentativas ESGOTADAS!
        Compras BLOQUEADAS'};
53     snackbarContainer.MaterialSnackbar.showSnackbar(data);
54     break;
55
56     case 'clean':
57     var snackbarContainer = document.querySelector('#toast-clear
        ');
58     var showToastButton = document.querySelector('#show-toast');
59     var data = {message: 'Item vazio'};
60     snackbarContainer.MaterialSnackbar.showSnackbar(data);
61     break;
62
63     default:
64     break;
65
66 }
67 }

```

---

Concluída a compra, é feito então o armazenamento de até cinco posições de compras na memória local. Para isso, utiliza-se as seguintes funções:

```

1  var latitude;
2  var longitude;
3  var id;
4
5  if (window.navigator && window.navigator.geolocation) {
6      var geolocation = window.navigator.geolocation;
7      geolocation.getCurrentPosition(sucesso, erro);
8  } else {
9      alert('Geolocalizacao nao suportada em seu navegador. ');
10 }
11 function sucesso(posicao){
12

```

```

13     latitude = posicao.coords.latitude;
14     longitude = posicao.coords.longitude;
15
16     cond_1 = window.localStorage.getItem("latitude_1");
17     cond_2 = window.localStorage.getItem("latitude_2");
18     cond_3 = window.localStorage.getItem("latitude_3");
19     cond_4 = window.localStorage.getItem("latitude_4");
20     cond_5 = window.localStorage.getItem("latitude_5");
21
22     id = 0;
23     // Verificar a Tag vazia para armazenamento
24     while (id != 1) {
25         if (cond_1 == null) {
26             window.localStorage.setItem("latitude_1", latitude);
27             window.localStorage.setItem("longitude_1", longitude
                );
28             tag = window.localStorage.getItem("tag_1");
29             id = 1;
30             document.getElementById("location_info").textContent
                = "Data : " + window.localStorage.getItem("
                    extrato_date_1") +
31             "\n\n Item: " + window.localStorage.getItem("
                    extrato_infos_1")+
32             "\n\n Total: R$ " + window.localStorage.getItem("
                    extrato_total_1");
33         }
34         else if (cond_2 == null) {
35             window.localStorage.setItem("latitude_2", latitude);
36             window.localStorage.setItem("longitude_2", longitude
                );
37             tag = window.localStorage.getItem("tag_2");
38             id = 1;
39             document.getElementById("location_info").textContent
                = window.localStorage.getItem("extrato_date_2")
                +
40             "\n - " + window.localStorage.getItem("
                    extrato_infos_2")+
41             "\nR$ " + window.localStorage.getItem("
                    extrato_total_2")+
42             "\n" + window.localStorage.getItem("tag_2");
43         }
44         else if (cond_3 == null) {
45             window.localStorage.setItem("latitude_3", latitude);
46             window.localStorage.setItem("longitude_3", longitude
                );
47             tag = window.localStorage.getItem("tag_3");
48             id = 1;

```

```

49         document.getElementById("location_info").textContent
           = window.localStorage.getItem("extrato_date_3")
             +
50         "\n - " + window.localStorage.getItem("
           extrato_infos_3")+
51         "\nR$ " + window.localStorage.getItem("
           extrato_total_3")+
52         "\n" + window.localStorage.getItem("tag_3");
53     }
54     else if (cond_4 == null) {
55         window.localStorage.setItem("latitude_4", latitude);
56         window.localStorage.setItem("longitude_4", longitude
           );
57         tag = window.localStorage.getItem("tag_4");
58         id = 1;
59         document.getElementById("location_info").textContent
           = window.localStorage.getItem("extrato_date_4")
             +
60         "\n - " + window.localStorage.getItem("
           extrato_infos_4")+
61         "\nR$ " + window.localStorage.getItem("
           extrato_total_4")+
62         "\n" + window.localStorage.getItem("tag_4");
63     }
64     else if (cond_5 == null) {
65         window.localStorage.setItem("latitude_5", latitude);
66         window.localStorage.setItem("longitude_5", longitude
           );
67         tag = window.localStorage.getItem("tag_5");
68         id = 1;
69         document.getElementById("location_info").textContent
           = window.localStorage.getItem("extrato_date_5")
             +
70         "\n - " + window.localStorage.getItem("
           extrato_infos_5")+
71         "\nR$ " + window.localStorage.getItem("
           extrato_total_5")+
72         "\n" + window.localStorage.getItem("tag_5");
73     }
74     else{
75         // Lotou
76         id = 1;
77     }
78 }
79
80 var myPosition = {lat: latitude, lng: longitude};
81

```



```

82         // Create a map object and specify the DOM element for
           display.
83         var map = new google.maps.Map(document.getElementById('map')
           , {
84             center: myPosition,
85             scrollwheel: false,
86             zoom: 16
87         });
88
89         // Create a marker and set its position.
90         marker = new google.maps.Marker({
91             map: map,
92             icon: image,
93             position: myPosition,
94             title: 'Hello World!',
95             label: tag,
96             draggable: true,
97             animation: google.maps.Animation.BOUNCE
98         });
99     }
100
101     function erro(error){
102         alert(error);
103     }
104     // Animando para o marcador pular
105     function toggleBounce() {
106         if (marker.getAnimation() !== null) {
107             marker.setAnimation(null);
108         } else {
109             marker.setAnimation(google.maps.Animation.BOUNCE);
110         }
111     }

```

---

### 5.3 Informações de compra

Para exibição das informações de compra na tela de Cartão, é utilizado o seguinte arquivo: *card.js*. Nele são carregadas as informações de compras realizadas e atribuídas a tabelas: uma para exibição de extrato e a outra para envio no formato de email.

```

1  var numero_cartao = window.localStorage.setItem("numero_cartao", '
           XXXX XXXX XXXX 2280'); // '9999 8902 1464 2280'
2  var limite_inicial = window.localStorage.setItem("limite_inicial",
           7000);
3  var tentativas = window.localStorage.setItem("tentativas", 3);
4

```

```

5 // Infos do Extrato - Compra 1
6 var name_and_qtd_1 = window.localStorage.getItem("extrato_infos_1");
7 var total_1 = window.localStorage.getItem("extrato_total_1");
8 var total_1_text = "R$ " + window.localStorage.getItem("
    extrato_total_1");
9 var date_1 = window.localStorage.getItem("extrato_date_1");
10
11 // Infos do Extrato - Compra 2
12 var name_and_qtd_2 = window.localStorage.getItem("extrato_infos_2");
13 var total_2 = window.localStorage.getItem("extrato_total_2");
14 var total_2_text = "R$ " + window.localStorage.getItem("
    extrato_total_2");
15 var date_2 = window.localStorage.getItem("extrato_date_2");
16
17 // Infos do Extrato - Compra 3
18 var name_and_qtd_3 = window.localStorage.getItem("extrato_infos_3");
19 var total_3 = window.localStorage.getItem("extrato_total_3");
20 var total_3_text = "R$ " + window.localStorage.getItem("
    extrato_total_3");
21 var date_3 = window.localStorage.getItem("extrato_date_3");
22
23 // Infos do Extrato - Compra 4
24 var name_and_qtd_4 = window.localStorage.getItem("extrato_infos_4");
25 var total_4 = window.localStorage.getItem("extrato_total_4");
26 var total_4_text = "R$ " + window.localStorage.getItem("
    extrato_total_4");
27 var date_4 = window.localStorage.getItem("extrato_date_4");
28
29 // Infos do Extrato - Compra 5
30 var name_and_qtd_5 = window.localStorage.getItem("extrato_infos_5");
31 var total_5 = window.localStorage.getItem("extrato_total_5");
32 var total_5_text = "R$ " + window.localStorage.getItem("
    extrato_total_5");
33 var date_5 = window.localStorage.getItem("extrato_date_5");
34
35 // Verificando valores
36
37 // Verificando se sao nulos
38 if (total_1 == null) {
39     name_and_qtd_1 = '';
40     total_1 = 0;
41     total_1_text = '';
42     date_1 = '';
43 }
44 if (total_2 == null) {
45     name_and_qtd_2 = '';
46     total_2 = 0;

```

```

47     total_2_text = '';
48     date_2 = '';
49 }
50 if (total_3 == null) {
51     name_and_qtd_3 = '';
52     total_3 = 0;
53     total_3_text = '';
54     date_3 = '';
55 }
56 if (total_4 == null) {
57     name_and_qtd_4 = '';
58     total_4 = 0;
59     total_4_text = '';
60     date_4 = '';
61 }
62 if (total_5 == null) {
63     name_and_qtd_5 = '';
64     total_5 = 0;
65     total_5_text = '';
66     date_5 = '';
67 }
68
69 // Somando os valores
70 var total_geral = parseInt(total_1) + parseInt(total_2) + parseInt
    (total_3)
71     + parseInt(total_4) + parseInt(total_5);
72
73 // Verificando o limite
74
75 if (total_geral != 0) {
76     var result = (window.localStorage.getItem("limite_inicial") -
        total_geral);
77     window.localStorage.setItem("limite_atual",result);
78     var limite_atual = window.localStorage.getItem("limite_atual");
79 } else {
80     window.localStorage.setItem("limite_atual",
        window.localStorage.getItem("limite_inicial"));
81     var limite_atual = window.localStorage.getItem("limite_atual");
82 }
83
84 // Criando tabela
85 function criarTabela(conteudo) {
86     var tabela = document.createElement("table");
87     var thead = document.createElement("thead");
88     var tbody = document.createElement("tbody");
89     var thd = function (i) { return (i == 0) ? "th" : "td"; };
90     for (var i = 0; i < conteudo.length; i++) {

```

```

91     var tr = document.createElement("tr");
92     for (var o = 0; o < conteudo[i].length; o++) {
93         var t = document.createElement(thd(i));
94         var texto = document.createTextNode(conteudo[i][o]);
95         t.appendChild(texto);
96         tr.appendChild(t);
97     }
98     (i == 0) ? thead.appendChild(tr) : tbody.appendChild(tr);
99 }
100 tabela.appendChild(thead);
101 tabela.appendChild(tbody);
102 return tabela;
103 }
104
105 function cardDefinitions(){
106     // Exibir numero do cartao e limite
107     document.getElementById("txtNumeroCartao").textContent =
108         window.localStorage.getItem("numero_cartao");
109     document.getElementById("txtLimite").textContent = "R$ " +
110         limite_atual + ",00";
111     document.getElementById("tabela").appendChild(criarTabela([
112         ["Data/Hora", "Descricao", "Valor"],
113         [date_1, name_and_qtd_1, total_1_text],
114         [date_2, name_and_qtd_2, total_2_text],
115         [date_3, name_and_qtd_3, total_3_text],
116         [date_4, name_and_qtd_4, total_4_text],
117         [date_5, name_and_qtd_5, total_5_text]
118     ]));
119 }
120
121 // Envio por email //
122 function sendEmail() {
123     var email = "altairpet@gmail.com";
124     var date = new Date();
125     var data = date.toLocaleDateString() + " " +
126         date.toLocaleTimeString();
127     var assunto = "Fatura solicitada - " + data;
128     var mensagem = "
129         Data/Hora | Descricao |
130         Valor " +
131         "\n" + date_1 + " " + name_and_qtd_1 + " " + total_1_text +
132         "\n" + date_2 + " " + name_and_qtd_2 + " " + total_2_text +
133         "\n" + date_3 + " " + name_and_qtd_3 + " " + total_3_text +
134         "\n" + date_4 + " " + name_and_qtd_4 + " " + total_4_text +
135         "\n" + date_5 + " " + name_and_qtd_5 + " " + total_5_text +
136         "\n\n Total a ser pago: R$" + total_geral;

```

```

133     window.location.href = "mailto:" + email + "?Subject=" + assunto +
        "&Body=" + mensagem + "\n";
134     //alert("Fatura enviada para: " + email + " na data: " + data + "
        com sucesso!" + "\n\n" + mensagem);
135     //window.document.getElementById("fatura").textContent = "Para: "
        + email + "na data: " + data + " com sucesso!" + mensagem;
136     window.localStorage.clear();
137     //window.location.href = "card.html";
138 }

```

---

A tabela principal que exibe o extrato é carregada através do HTML no id container da seguinte forma:

```

1 <div class="container">
2   <div class="card-3"></div>
3   <div class="container" id="tabela">
4     <h5>Extrato</h5>
5     <div id="container"><!--Tabela--></div>
6   </div>
7 </div>

```

---

Para exibição de posição do usuário no momento da compra no mapa, é utilizado o arquivo *maps.load.js* que carrega as informações armazenadas na memória e exibe. O posicionamento é feito através da leitura de latitude e longitude armazenada para respectiva compra.

```

1 var image = 'images/localization/marker.png';
2 var marker_content = window.document.getElementById('
    marker-dialog-content');
3 var marker_dialog = document.getElementById('marker-dialog');
4
5 var cond_1 = window.localStorage.getItem("latitude_1");
6 var cond_2 = window.localStorage.getItem("latitude_2");
7 var cond_3 = window.localStorage.getItem("latitude_3");
8 var cond_4 = window.localStorage.getItem("latitude_4");
9 var cond_5 = window.localStorage.getItem("latitude_5");
10
11 var latitude_1; var latitude_2; var latitude_3; var latitude_4; var
    latitude_5;
12 var longitude_1; var longitude_2; var longitude_3; var longitude_4;
    var longitude_5;
13 var tag_1; var tag_2; var tag_3; var tag_4; var tag_5;
14 var marker_1; var marker_2; var marker_3; var marker_4; var
    marker_5;
15

```

```

16 var lat_init; var long_init;
17
18 if (window.navigator && window.navigator.geolocation) {
19     var geolocation = window.navigator.geolocation;
20     geolocation.getCurrentPosition(sucesso, erro);
21 } else {
22     alert('Geolocalizacao nao suportada em seu navegador.');
```

```

23 }
24 function sucesso(posicao) {
25
26     lat_init = posicao.coords.latitude;
27     long_init = posicao.coords.longitude;
28     var myPosition = {lat: lat_init, lng: long_init};
29
30     // Create a map object and specify the DOM element for
31     // display.
32     var map = new google.maps.Map(document.getElementById('map'
33     ), {
34         center: myPosition,
35         scrollwheel: false,
36         zoom: 16
37     });
38     // Verificando Tags vazias
39     if (cond_1 != null) {
40         tag_1 = window.localStorage.getItem("tag_1");
41         latitude_1 = parseFloat(window.localStorage.getItem("
42             latitude_1"));
43         longitude_1 = parseFloat(window.localStorage.getItem("
44             longitude_1"));
45
46         // Marker 1
47         var myPosition_1 = { lat: latitude_1, lng: longitude_1
48             };
49
50         marker_1 = new google.maps.Marker({
51             map: map,
52             icon: image,
53             position: myPosition_1,
54             label: tag_1,
55             draggable: false,
56             animation: google.maps.Animation.BOUNCE
57         });
58         marker_1.addListener('click', toggleBounce);
59
60         function toggleBounce() {
61             if (marker_1.getAnimation() !== null) {
62                 marker_1.setAnimation(null);

```

```

58         } else {
59             marker_1.setAnimation(
60                 google.maps.Animation.BOUNCE);
61         }
62     marker_1.addListener('click', showInfos_1);
63
64 }
65 if (cond_2 != null) {
66     tag_2 = window.localStorage.getItem("tag_2");
67     latitude_2 = parseFloat(window.localStorage.getItem("
68         latitude_2"));
69     longitude_2 = parseFloat(window.localStorage.getItem("
70         longitude_2"));
71
72     // Marker 2
73     var myPosition_2 = { lat: latitude_2, lng: longitude_2
74         };
75
76     marker_2 = new google.maps.Marker({
77         map: map,
78         icon: image,
79         position: myPosition_2,
80         label: tag_2,
81         draggable: false,
82         animation: google.maps.Animation.BOUNCE
83     });
84     marker_2.addListener('click', toggleBounce);
85     function toggleBounce() {
86         if (marker_2.getAnimation() !== null) {
87             marker_2.setAnimation(null);
88         } else {
89             marker_2.setAnimation(
90                 google.maps.Animation.BOUNCE);
91         }
92     }
93     marker_2.addListener('click', showInfos_2);
94
95 }
96 if (cond_3 != null) {
97     tag_3 = window.localStorage.getItem("tag_3");
98     latitude_3 = parseFloat(window.localStorage.getItem("
99         latitude_3"));
100     longitude_3 = parseFloat(window.localStorage.getItem("
101         longitude_3"));
102
103     // Marker 3

```

```

98         var myPosition_3 = { lat: latitude_3, lng: longitude_3
99             };
100
101     marker_3 = new google.maps.Marker({
102         map: map,
103         icon: image,
104         position: myPosition_3,
105         label: tag_3,
106         draggable: false,
107         animation: google.maps.Animation.BOUNCE
108     });
109     marker_3.addListener('click', toggleBounce);
110     function toggleBounce() {
111         if (marker_3.getAnimation() !== null) {
112             marker_3.setAnimation(null);
113         } else {
114             marker_3.setAnimation(
115                 google.maps.Animation.BOUNCE);
116         }
117     }
118     marker_3.addListener('click', showInfos_3);
119 }
120 if (cond_4 != null) {
121     tag_4 = window.localStorage.getItem("tag_4");
122     latitude_4 = parseFloat(window.localStorage.getItem("
123         latitude_4"));
124     longitude_4 = parseFloat(window.localStorage.getItem("
125         longitude_4"));
126
127     // Marker 4
128     var myPosition_4 = { lat: latitude_4, lng: longitude_4
129         };
130
131     marker_4 = new google.maps.Marker({
132         map: map,
133         icon: image,
134         position: myPosition_4,
135         label: tag_4,
136         draggable: false,
137         animation: google.maps.Animation.BOUNCE
138     });
139     marker_4.addListener('click', toggleBounce);
140     function toggleBounce() {
141         if (marker_4.getAnimation() !== null) {
142             marker_4.setAnimation(null);
143         } else {
144             marker_4.setAnimation(

```



```

                                google.maps.Animation.BOUNCE);
140         }
141     }
142     marker_4.addListener('click', showInfos_4);
143 }
144 if (cond_5 != null) {
145     tag_5 = window.localStorage.getItem("tag_5");
146     latitude_5 = parseFloat(window.localStorage.getItem("
        latitude_5"));
147     longitude_5 = parseFloat(window.localStorage.getItem("
        longitude_5"));
148
149     // Marker 5
150     var myPosition_5 = { lat: latitude_5, lng: longitude_5
        };
151
152     marker_5 = new google.maps.Marker({
153         map: map,
154         icon: image,
155         position: myPosition_5,
156         label: tag_5,
157         draggable: false,
158         animation: google.maps.Animation.BOUNCE
159     });
160     marker_5.addListener('click', toggleBounce);
161     function toggleBounce() {
162         if (marker_5.getAnimation() !== null) {
163             marker_5.setAnimation(null);
164         } else {
165             marker_5.setAnimation(
                google.maps.Animation.BOUNCE);
166         }
167     }
168     marker_5.addListener('click', showInfos_5);
169 }
170 }
171 function erro(error){
172     alert(error);
173 }
174
175 function showInfos_1(){
176
177     var date_1 = window.localStorage.getItem("extrato_date_1");
178     var name_and_qtd_1 = window.localStorage.getItem("
        extrato_infos_1");
179     var total_1_text = "R$ " + window.localStorage.getItem("
        extrato_total_1");

```

```

180
181         if (! marker_dialog.showModal) {
182             dialogPolyfill.registerDialog(marker_dialog);
183         }
184         marker_dialog.showModal();
185         marker_content.textContent = "Data: " + date_1 + "\
            nProdutos: " + name_and_qtd_1 + " \nTotal: " +
            total_1_text;
186         marker_dialog.querySelector('.close').addEventListener('
            click',function(){
187             marker_dialog.close();
188         });
189     }
190     function showInfos_2(){
191
192         var date_2 = window.localStorage.getItem("extrato_date_2");
193         var name_and_qtd_2 = window.localStorage.getItem("
            extrato_infos_2");
194         var total_2_text = "R$ " + window.localStorage.getItem("
            extrato_total_2");
195
196         if (! marker_dialog.showModal) {
197             dialogPolyfill.registerDialog(marker_dialog);
198         }
199         marker_dialog.showModal();
200         marker_content.textContent = "Data: " + date_2 + "\
            nProdutos: " + name_and_qtd_2 + " \nTotal: " +
            total_2_text;
201         marker_dialog.querySelector('.close').addEventListener('
            click',function(){
202             marker_dialog.close();
203         });
204     }
205     function showInfos_3(){
206
207         var date_3 = window.localStorage.getItem("extrato_date_3");
208         var name_and_qtd_3 = window.localStorage.getItem("
            extrato_infos_3");
209         var total_3_text = "R$ " + window.localStorage.getItem("
            extrato_total_3");
210
211         if (! marker_dialog.showModal) {
212             dialogPolyfill.registerDialog(marker_dialog);
213         }
214         marker_dialog.showModal();
215         marker_content.textContent = "Data: " + date_3 + "\
            nProdutos: " + name_and_qtd_3 + " \nTotal: " +

```

```

        total_3_text;
216         marker_dialog.querySelector('.close').addEventListener('
            click',function(){
217             marker_dialog.close();
218         });
219     }
220     function showInfos_4(){
221
222         var date_4 = window.localStorage.getItem("extrato_date_4");
223         var name_and_qtd_4 = window.localStorage.getItem("
            extrato_infos_4");
224         var total_4_text = "R$ " + window.localStorage.getItem("
            extrato_total_4");
225
226         if (! marker_dialog.showModal) {
227             dialogPolyfill.registerDialog(marker_dialog);
228         }
229         marker_dialog.showModal();
230         marker_content.textContent = "Data: " + date_4 + "\
            nProdutos: " + name_and_qtd_4 + " \nTotal: " +
            total_4_text;
231         marker_dialog.querySelector('.close').addEventListener('
            click',function(){
232             marker_dialog.close();
233         });
234     }
235     function showInfos_5(){
236
237         var date_5 = window.localStorage.getItem("extrato_date_5");
238         var name_and_qtd_5 = window.localStorage.getItem("
            extrato_infos_5");
239         var total_5_text = "R$ " + window.localStorage.getItem("
            extrato_total_5");
240
241         if (! marker_dialog.showModal) {
242             dialogPolyfill.registerDialog(marker_dialog);
243         }
244         marker_dialog.showModal();
245         marker_content.textContent = "Data: " + date_5 + "\
            nProdutos: " + name_and_qtd_5 + " \nTotal: " +
            total_5_text;
246         marker_dialog.querySelector('.close').addEventListener('
            click',function(){
247             marker_dialog.close();
248         });
249     }

```

---